```
User name:   SHAW (11)                      Queue:  UIDESK/UNIX_UIMKGT
File name:                                  Server: CLFPCD010
Directory:
Description: NORDIC29.
             December 21, 1998                      5:38pm

             SSS  H    H    A    W    W
            S   S H    H   A A   W    W
            S     H    H  A   A  W    W
             SSS  HHHHHH  A   A  W  W W
                S H    H  AAAAA  W  W W
            S   S H    H  A   A  WW  WW
             SSS  H    H  A   A  W    W

            L       SSS   TTTTT
            L      S   S    T
            L      S        T     ::
            L       SSS     T     ::
            L          S    T
            L      S   S    T     ::
            LLLLL   SSS     T     ::
```

EXHIBIT

RX-581 C

Cirrus Confidential
Business Information

CL 152684

BEST AVAILABLE COPY

NORDIC-1M Design Specification

by Vlad Bril, Robin Han, Sasha Eglit, Rakesh Bindlish, Dwarka Partani.
Jeff Ort
rev.5.2 December 21, 1998

1.0 Introduction & Table of Contents

Nordic is Cirrus Logic's Fourth Generation Flat Panel VGA Controller to be marketted in first half of 1994.
Nordic-1M is a mid-end member of the Nordic family. It is a GUI Accellerator with PCI support and Multi-Media Features.
Nordic-1M is targetted for the mid to high end of the **Note-book Computer Market** and for for the energy efficient **desk-top computers mother-boards**.
Nordic-1M is **not** targetted to the adapter market.

1.1 Table of Contents

This Specification will address the following:

2.0 Nordic-1M Basic Feature Set

1. **Flat Panel VGA Compatible GUI Accellerator register and feature compatible to 5428 and Alpine I A/V. Priority: A (= highest).**
2. **32bit CPU, 32bit Video Memory, 32bit BLT and 32bit CPU to Video Memory Data Path. Priority: A (= highest).**
3. **208 pin QFP package.**
4. **Targeted to sample CY Q1'94, production CY Q2 '94. Priority: A (= highest).**
5. **Target cost: $12 to $10 -> less than 400mil [/] in C8 Foster City.**

6. **Multimedia Features**

- Part of a well defined **System Solution** for high performance **CD-ROM video and audio playback** for the portable market. **Priority: B**
- Part of a well defined **System Solution** for mid performance **low cost live NTSC Video and Audio with the NTSC decoder on a PCMCIA card. Priority: B**
- ~~**Audio Port to Display Memory**: Audio Buffer to Serial Interface for CS4248 Audio Codec. **Priority: C** (CS4215 is not fit for notebook computers).~~
- **Video Playback decompression accelleration for CinePak** including Color Space Conversion (YUV -> RGB). **Priority: B**
- **Live Video in a 320x240 or 640x480 Window** from a proprietaryly compressed TV decoded data source (at 30Hz) on TFT color panels. Data is stored compressed and it is decompressed to 18bit/pel or 24bit/pel in real time at display time. NTSC decoder and data compressor is on the PCMCIA card or the PCMCIA controller. Please note that due to live video bandwidth requirements on PCMCIA bus, a relatively non-expensive way to achieve the second feature is to implement this compression algorithm. **Priority: B**
- Color Key Live Video Overlay based on a 5428-like Feature Connector with PCI bus (only).

7. **Integrated in Nordic 1-M**

Cirrus Confidential Business Information

CL 152686

- Color Space Converter for CinePak. **Priority: B**
- 18bit DAC with direct color capability **Priority: A**
- *24bit RAMDAC with true color capability* **Priority: C**
- Clock Synthesizers for Video and Memory Clock (65MHz/3V, 110MHz/4.5V). **Priority: A**

- DAC IRef Current Source on chip. **Priority: C.** *Boyd and ManShek are working on it*

8. **Direct Bus Interface**:

- 32bit VESA-VL bus (and 486 local bus) **Priority: A**
- 32bit PCI bus. No on chip BIOS ROM support in PCI. **Priority:** A
- 16bit ISA bus for demonstration and easy system validation. **Priority: B**
- 32K or 48KB BIOS ROM support in ISA and VESA VL bus. **Priority: B** (only for BIOS/ system debug)
- At least 5 scratch pad registers: optimum 8 scratch pad registers. **Priority: A.**

9. **Performance Enhancements** to achieve 15-20MWinmarks at 1Kx768 8b/pel:

- 32bit BitBLT with memory mapped I/O BitBLT registers (as in 5428 ?or Alpine) Priority:A
- BLT buffer of 8 (~~or 16~~) bytes for Source (and Destination ?) data. (5428 buffer is 8bytes. Alpine is 16bytes) (as 5428)
- *X/Y offset for Pattern BLT (as in Alpine)* **Priority: C**
- h/w graphic cursor 64x64 (as in 5428)
- color expansion (as in 5428)
- linear memory addressing (as in 5428)
- 4 stage 32bit wide write buffer and capability to page CPU cycles if accumulated in the write buffer (this is already in 5428)

10. **Display Memory:**

- *0.5MB (16bit wide)*, 1MB or 2MB memory addressing 32bit wide (as in 5428)
- *1 or* 2 256Kx16 one bank or 4 256Kx16 two banks (2 CAS symetric, 2 WE asymetric) (as in 5428)
- maximum memory clock frequency: 50MHz/3V, 65MHz/4.5V **Priority: C**

11. **Flat Panel Support**: **[Priority: A]**

- Color TFT (24bit, 18bit, 12bit, 9bit)
- Color Dual Scan and *Single Scan STN* (16 and 8 bit, including dual clock panels)
- Monochrome TFT (8bit),
- Monochrome Dual Scan STN (8bit),
- Monochrome Single Scan (4 and 8bit) support

- Simulscan with all Fast Panels (6.3MHz or equivalent) with 32bit wide memory.
  No simulscan with 16bit wide memory with dual scan color panels.
  No simulscan for 3MHz monochrome dual scan panels.
- 640x480 single and dual scan support (Priority:A)
- 800x600 TFT *and no single and dual mono and color STN panel support (Priority: C)*
- 1024x768 TFT *and no single and dual scan STN panel support . (Priority: D)*

12.Display Features

- High quality monochrome and color shading, at least as good as 6440. Priority: A
- Shading option for fast panels. Priority: B
- Up to 8, 64x64 Hardware Icons independently mapped color+transparent background. **Priority: C**
- *Whisper Cross-Talk Reduction (If panels will be available in the time frame and if we can accomodate it with the shading algorithm).* ***Priority: C***
- *"**Office Productivity Booster Dual Display**"* 640x480 (panel) & 640x480 (CRT) 4 or 8b/pel or 640x480 (panel) & 1024x768 (CRT) 4 or 8b/pel ***Priority: E***
- Vertical Expansion in text and graphics (as in 6235 or 6440)
- Automatic Centering if not expanded (as in 6235)
- *Grey-Shades Maping options ?? Is this needed ? Priority: D*
- Text Contrast Enhancement options. (as in 6235 or 6440)
- *Inking Plane ?? Priority: D*
- *180 and 90 degree image rotation ?? Priority: D*
- *12 and 16dots wide font support ?? Priority: D*
- *h/w offset on h/w cursor ?? Priority: D*

13.Resolutions Supported:

- the same as 5428 at 5V (4.5Vmin)
- up-to 45MHz dotclk (1024x768, 60Hz interlaced) at 3.3V

Cirrus Confidential
Business Information

14.Power Management

- Mixed 3.3V and 5V operation: memory, host and flat panel independent VDD
- Green-PC spec support. **Priority: B**
- Stand-by Mode with internal stand-by timer (as in 6235)

CL 152688

- Suspend Mode I/O or pin in all modes (LCD. CRT.Simulscan)
- *Light Sleep ?? - better not for this chip as it requires too much h/w. Priority: F*
- Panel Power Sequencing: automatic or under I/O control (as in 6235)
- Reduced Active Power in Panel Only Operation: < 60mAmp @3.3V. **Priority:A**
- Reduced Suspend Current: < 0.5mAmp. **Priority: A**

2.1 Inputs from the Nordic-1M Features Meeting of 09/10/93

- 24bit DAC if not too expensive (VAR)
- mclk @5V max 65MHz. @3V max 50MHz
- vclk and fpvdclk @5V max 85MHz, @3V max 65MHz
- **panels:**
    - TFT 1280x1024? ##, **1kx768 ##, 800x600 ##**, 640x480
      24bit,18bit, 12bit, 9bit color, 6bit mono
      NEC analog output? ##
    - STN 800x600 dual scan color, 640x480 ds color 16bit
      640x480 ss color 8bit
      640x480 mono bit 8bit and 4bit
    - horiz ## and vertical centering
    - normal 9dot font in 800x600 text ##
    - simulscan for 800x600 and 1kx768 panels ## (need to run all modes at 800x600 or 1024x768 rezolutions -> option to clock panel till h&v Blank)
    - support for two alternate fonts 10x24 for 800x600 (10x18 actual with spacing) and 12x25 for 1kx768 ## difficult !!
    - h&v text expansion in 800x600 and 1kx768 panels ## ?
    - color text enhancement ## : bright white instead of white
    - *on chip cross-talk reduction ##* ? - any improvement ?
    - *shading look-up tables ## ?* - only if not too large
- ***NO Whisper support even on the bus side - out !***
- ***NO one 256Kx16 DRAM support - out !***
- **h/w icon support:** with h/w cursor, up to 4 simultaneously. Horizontal position controlled in increments of 64 pixels.
  Icon attributes: off/on, 4 colors/3colors + transparent, no_blink/blink, (simple/double).
  H/W cursor goes over the h/w icon (visually). Uses independenly mapped color in RAMDAC RAM. Start at scan-line 64.
- PC98 integration? ## - to be negotiated
- reserve pins for SDRAM-s and CDRAM-s ##
- ***NO true dual display!***

Cirrus Confidential
Business Information

2.2 Inputs from Compaq 09/10/93

CL 152689

- 50MHz 486DX bus
- 75Hz refresh rate
- **need one 256Kx16 support with Audio and some video for Contura - build one motherboard for Contura and LT Lite. Two DRAM-s only on LT Lite.**

- think that 8KB/buffer of audio buffer is not enough
- want to use Analog Devices for Audio
- No clear winner on conpression. They are watching. Need Indeo driver.
- Play-back zoom is required. At least pixel replication.
- Very interested in Compressed Live Video. WAnt to talk more.
- need h/w icon 64x64 with 3col+transparency or 4 color, with pixel replication to 128x128.

2.3 Inputs from AST 09/15/93
- They prefer horizontal icon

2.4 New Features as of 10/04/93
- 4bit/pel packed support for Windows Chicago
- 8bit/pel video out to the Feature Connector for conversion to NTSC-out with Nordic running 640x480 interlaced mode.
  Another idea was to use some of the panel outputs in this case and extend the data out to 16bits.
- The h/w icon can be horizontal or vertical using the Motion Video Window as implementation vechicle -> is it worth?
- IBM Raleigh asked for 12 and 16dots wide fonts for h/w assisted Kanji. 10/04/93
- IBM Raleigh asked for 8bit monochrome TFT panel support. 10/04/93

3.0 Important Additions to 5428 Data Base

1. **32 bit CPU Data bus in VESA VL bus support**
2. **PCI bus support**
3. **Mono and color, STN and TFT Flat Panel support (incl. hfb).**
4. **Live video in a true color window with Windows running in planar VGA mode, without a video port (using PCMCIA standard bus and Sasha's decompression technique).**
5. **CINE PAK decompression accelleration**
6. **Audio Port support for CS4215**
7. **Power Management: panel power sequencing, stand-by mode, suspend mode**
8. **Mixed Voltage, 3.3V operation and low active/suspend/standby power**
9. **h/w icon support, with a h/w cursor support**
10. **BLT performance improvements: Memory Mapped BLT I/O, patternning, 16 byte Source buffer**

11.*[dual display 1024x768 on CRT with 640x480 on LCD, using the same RAMDAC] - if decided to do now*

4.0 Detailed List of Features

4.1 WAS: On the Cross Talk Reduction Technique -> not in - 1M

This part was deleted from Nordic-1M spec starting rev. 3.8 (09/11/93).

4.2 Green PC Spec Support (Based on VESA DPMS Proposal 1.0p rev. 0.53p)

This specification refferes to CRT Monitor power management. Additionally, Nordic will

TABLE 1. Display Power Management Summary

State	HSYNC	VSYNC	Video RGB	Compliance	Power Savings	Recovery Time
CRT On	Pulsese	Pulsees	Active	Mandatory	None	NA
CRT Stand-by	No Pulses	Pulses	Blanked	Optional	Minimal	Short
CRT Suspend	Pulses	No Pulses	Blanked	Mandatory	Substantial	Longer
CRT Off	No Pulses	No Pulses	Blanked	Mandatory	Maximum	System Dependednt

control DAC power consumption. These power management CRT Monitor states are not necessarily related to the graphics controller power management states.

We'll define 2 bits - GRE[2:1] (the same as Alpine) that control HSYNC, VSYNC and DAC Power Off signal (or Blankn to the DAC).

TABLE 2. GRE[2:1] Effect

GRE[2:1]	CRT State	Vsync	Hsync	DAC Power	Nordic Power Management State
0	On	pulsing	pulsing	on	active
1	Stand-by	pulsing	static at MISC[6] inactive level	off	active
2	Suspend	static at MISC[7] inactive level	pulsing	off	active or stand-by
3	Off	static at MISC[7] inactive level	static at MISC[6] inactive level	off	active, stand-by or suspend

If Nordic is in its own Suspend mode, the BIOS should program GRE[2:1]=3 to put the monitor (if any) in the Off state.

If Nordic is in its own Stand-by mode, CRT Suspend mode should be forced bt the h/w, as stand-by is a timer driven mode and no s/w gets to play.

If Nordic is in LCD only, the BIOS should set GRE[2:1] to 3 to power off the CRT that may be connected to the notebook computer.

If CRT only mode and CRT is in stand-by or suspend, the BIOS may reduce the video clock and even memory clock frequencies after saving the current values.

Except for the above restrictions, CRT Power Management modes are not related to Nordic power management modes: they can be independently programmed.

If MISC bit is programmed for positive polarity, the SYNC signal will be stopped L, otherwise it will be stopped H.

4.3 Play-back Decompression Accelleration (applicable to Cinepak as well as other compression standards) - old, kept here only for reference

4.3.0 The problem

a) When playback data is compressed on a CD ROM, it is encoded in 24bit/pel YUV with crominance sub-sampling. Due to CDROM reduced bandwidth the display window is 160x120 or 320x240. If this compressed data is decompressed at full pixel depth, it should be displayed as 24bit/pel RGB window under MS Windows. As the play-back data is sent to the graphics controller from the CPU bus, it has to be placed in Video Memory in 320KB.

A standard VGA controller or even GD5428 are able to display the 24bit/pel play-back window only if MS Windows is run in 24bit/pel mode (for instance 1024x768 24bit/pel or 640x480 24bit/pel) by the driver. Under these conditions, the CPU will have to write the un-compressed 24bit/pel play-back data in the appropriate position in Video Memory, contigouus to the entire displayed data.
So, in order to display 24bit/pel in a play-back window, GD5428 needs the driver to run 24bit/pel at the maximum display resolution.
But, in order to run 640x480 at 24bit/pel we need 921.6KB of Video Memory and in order to run 1024x768 24bit/pel we need 2359.296KB of video memory. (Please note that 2M of memory has 2097.152KB).
Also, when running windows in 24bit/pel mode the performance will decrease 3 or 4 times relative to 8bit/pel mode, especially when using DRAM as Video Memory.

b) When running through MS Windows GDI, the driver can play-back data at 15fps or less (with a 486 33MHz). It is desirable to play-back data at 30fps.
This can be accomplished today only if the driver does not go through GDI.

c) If a picyure was stored on CDROM at low resolution (160x120 or 320x240) and it needs to be "zoomed", it has to be interpolated both horizontally and vertically for good picture quality. Vertical interpolation is expensive in h/w.

4.3.1 The Generic Solution - old, here for reference only

Today's Windows Accellerators run MS Windows, including the play-back window with the same pixel depth (8bit/pel), normally going through the RAMDAC, but using a split palette approach with 16 colors reserved for MS Windows and 240 colors reserved for the play-back window. The Cinepak driver writes data in video memory in 3:3:2 RGB format

(8bits/pel) which is created by the driver. Because of the s/w overhead involved in decompression as well as the GDI overhead, it is hard to go beyond 320x240 clips at 15fps, though the display quality is much lower than what Cinepak could achieve (240 colors versus 16 million colors).
To run Cinepak with 24pit/pel resolution, today's graphics controllers would require to run MS Windows in 24bit/pel mode, requiring almost 1MB in 640x480 and over 2MB-s in 1024x768 modes.

A graphics controller that could run MS Windows in 8 or 16bit/pel mode while displaying only the play-back window in 24bit/pel mode would increase significantly picture quality without degrading too much MS Windows performance with play-back. would reduce significantly the amount of Video Memory required for the same picture quality and would allow a higher frame rate for the play-back.

Further. if the play-back data is stored in Video Memory in a semi-compressed format that requires less than 24bit/pel this would further decrease the Video Memory requirements and it may even reduce the video memory band-width requirements.

When zooming the play-back window from 160x120 to 320x240 or higher it is important to interpolate. Vertical interpolation can be done in s/w but with a relatively large overhead. It may be possible to do vertical interpolation on key frames. but there is no practical way of doing vertical interpolation on inter-frames either in s/w or with h/w assist on the CPU write side of the graphics controller. If s/w driven vertical interpolation on key frames is not a solution. then verical interpolation can be only achived in h/w on the CRT read side and this requires either one or two 16bit/pel line buffers or a big penalty on video memory bandwidth (the equivalent of reading 32 or 48bits/pel).

Horizontal interpolation can be done before play-back data is displayed. without s/w help.

Please note that there are multiple ways to solve the play-back problems. In Nordic-1M we'll try to strike a ballance between the complexity of the h/w solution and the quality of the play-back image.

To a large extent. the solution adopted in Nordic-1M will be generic. not tied to Cinepak. though it will be first applied to Cinepak.

4.3.2 Nordic-1M old Play-back Accelleration Solution -> canned. kept here for reference

a)Nordic-1M will run Cinepak at 1:1 scale exactly the same way as today's graphics controllers: in 8bit/pel MS Windows with 8bit/pel (3:3:2 RGB) play-back window.

b)Nordic-1M will run 1:2 scale (zoom) in 8bit/pel 1024x768 MS Windows with 24bit/pel (8:8:8 RGB) pixel depth in the play-back window. horizontally interpolated and vertically doubled. Play-back data will be stored in video memory as 16bit/pel (Y0 U01 Y1 U01).

c)Nordic-1M will run at 1:1 scale in 16bit/pel 1024x768 MS Windows with 24bit/pel pixel depth in the play-back window. Play-back data will be stored in video memory as 16bit/pel (Y0 U01 Y1 U01). This mode is good for programs that store data in 320x240 format.

Because today large play-back windows look bad. while small ones look OK. this solution will improve picture quality to a large extent.

Concerns:

Many play-back programs today don't even have a zoom option. To this extent, mode "b" may not be needed.

Mode c could be implemented today with 16bit/pel RGB (555) in s/w and the only improvement we provide is 24bit/pel capability and h/w color conversion (some level of accelleration). So mode "c" may not be needed as a special h/w.

If this is the case, maybe we do not need to do anything in h/w for Cinepak !

One idea: should we run 24bit/pel MS windows stored in 4:2:2 (Y0 U01 Y1 U01) format? This will allow us to run 24bit/pel MS Windows and use 16bit/pel memory space with no loss of quality. The windows driver could compress the data and the h/w could decompress it.

4.4 Nordic-1M Motion Video Architecture

Sasha Eglit, Rakesh Bindlish, Vlad Bril and Dave Keene are important contributors to the Motion Video Architecture definition.

4.4.0 The Problems to Solve & Generic Solutions

Nordic-1M is supposed to support CDROM play-back and live-video under Microsoft's Video for Windows. With all of today's VGA Compatible Graphics Controllers, the play-back picture pixel depth has to be the same as the surrownding MS Windows pixel depth. In other words, we have to run MS Windows in a 24bit/pel mode in order to display 24bit/pel in the play-back window.
The only way we can run today live video is from a Video Port (= Feature Connector). In this case we'll use an overlay or a color key to define the live video window.
Due to CDROM player, s/w interface complexity, CPU bus and Video Memory limitations, only small clips at 15fps or less can be playd-back today. So today we need a lot of Video Memory to run small clips in slow motion at low rezolution. Nordic-1M will try to use less Video Memory, increase the clips size, their resolution and speed.

Nordic-1M is supposed to support:
- accellerated play-back for all popular CDROM compression standards, especially Cinepak and Indeo
- live video from a PCMCIA card in a system with a standard PCMCIA host adapter, for a "Multimedia-Ready Notebook Computer"
- live video from a Feature Connector, preferably VESA Advanced Feature Connector compatible, for a Multimedia Notebook Computer with direct NTSC input (full mother-

board solution).

What will Nordic-1M bring to the Video plate:

A. Nordic-1M will break the dependency between the pixel depth of MS Windows and the pixel depth of the live-video / play-back window. By doing so, Nordic-1M will be able to run 24bit/pel LV/PB while running MS Windows in 8bit/pel or 4bit/pel (even planar). This will lead to high quality live-video and play-back, while minimizing Video Memory requirements.

B. Nordic-1M will further reduce Video Memory requirements as well as Video Memory Bandwidth requirements by storing data in compressed form: 4:2:2 YUV (16bit/pel), 4:1:1 YUV (12bit/pel) or Sashapak YUV (2.6bit/pel)

C. Nordic-1M will define one architecture and work-frame to run both play-back and live-video in a unitary way for both s/w and h/w.

D. Nordic-1M will support up to two active LV/PB windows at the same time.

E. By providing h/w YUV->RGB conversion and 1:2 zoom Nordic-1M will accellerate playback for most standards, but mostly for standards that allow us access to their decompressor like Cinepack (for whose decompressor we have a licence).

Nordic-1M will not provide Cinepak specific accelleration (custom BLT).

On the Live Video from Feature Connector (or Video Port)

Due to pin-out limitations, the VAFC implementation has to be limitted to 8 pins of data requiring external muxing of 16bit data to Nordic. Further, the VAFC solution will be restricted to 5428 type support: overlay and color key (no memory video port). The only additions to the 5428 support will be the internal YUV-> RGB converter able to accept sequential 4:2:2 YUV (16bit/pel in avarage: Y0,U,Y1,V) and display it as a 24bit/pel RGB in the overlay/color-key window. We may also support horizontal 1:2 zoom with avaraging. Nordic-1M will be back-wards compatible to 5428 and it will support also 16bit 5-5-5 RGB Sierra and 5-6-5 RGB XGA pixel formats. As in 5428, live video from the Feature Connector can be executed only from mode 5F (640x480 256 colors).

In the following we will refer mostly to the other two modes of operation: play-back and live video from a PCMCIA card, which we'll call compressed live video, because due to PCMCIA bus bandwidth limitations, we assume that video data received by Nordic is compressed with Sashapak (see Sashapak detailes further in this spec). Please note that due to its huge advantages, it makes sense to use Sashapak as a mother-board solution too. The main reason we continue to support the Feature Connector based live video is the availability of drivers for 5428 which will give Nordic an early start in the live video arena.

Video Memory Band-width Constraints

Please note that, even with a 32bit wide Video Memory, there are severe memory band-width limitations when running 16bit/pel and 24bit/pel graphics modes.

Assumming that Nordic-1M does 8 CRT fetches in a row and 1 CPU cycle, for a CRT or TFT panel, here are the minimum frequency requirements for memory clock frequency at different rezolutions and pixel depths:

24bit/pel:
R+7P+R=12m+14m=26m with 1 CPU cycle per CRT FIFO fill fetches data for 32B/3B=10pixels
R+7P = 6m+14m = 20m with no CPU cycle during non-display time
min mclk freq = 65MHz / 50MHz @ 640x480 -> Nordic-1M upper limit at 5V CVDD
= 104MHz / 80MHz @ 800x600 -> too high
= 169MHz / 133MHz @ 1024x768 60Hz -> too high
16bit/pel:
R+7P+R=12m+14m=26m / R+7P = 20m fetches data for 32B/2B=16pixels
min mclk freq = 40.6MHz / 36MHz @ 640x480 -> OK even at 3V
= 65MHz / 50MHz @ 800x600 -> Nordic-1M upper limit at 5V CVDD
= 105MHz / 83MHz @ 1024x768 60Hz -> too high
12bit/pel:
R+7P+R=12m+14m=26m / R+7P=20m fetches data for 32B/1.5B=21pixels
min mclk freq = 30.9MHz / 23MHz @ 640x480 -> OK even at 3V
= 49.5MHz / 38MHz @ 800x600 -> OK even at 3V
= 80.4MHz / 64MHz @ 1024x768 60Hz -> too high
8bit/pel:
R+7P+R=12m+14m=26m fetches data for 32B/1B=32pixels
min mclk freq = 20.3MHz @ 640x480 -> OK even at 3V
= 40.6MHz @ 800x600 -> OK even at 3V
= 52.8MHz @ 1024x768 60Hz -> OK at 5V

We may conclude that Nordic-1M will be able to run 24bit/pel only at 640x480 rezolution. 16bit/pel up to 800x600, 12bit/pel up to 800x600 and 8bit/pel up to 1024x768 rezolution. These results apply only to CRT or TFT and single scan STN panels.
Please note that for dual scan STN panels the memory requirements increase significantly.

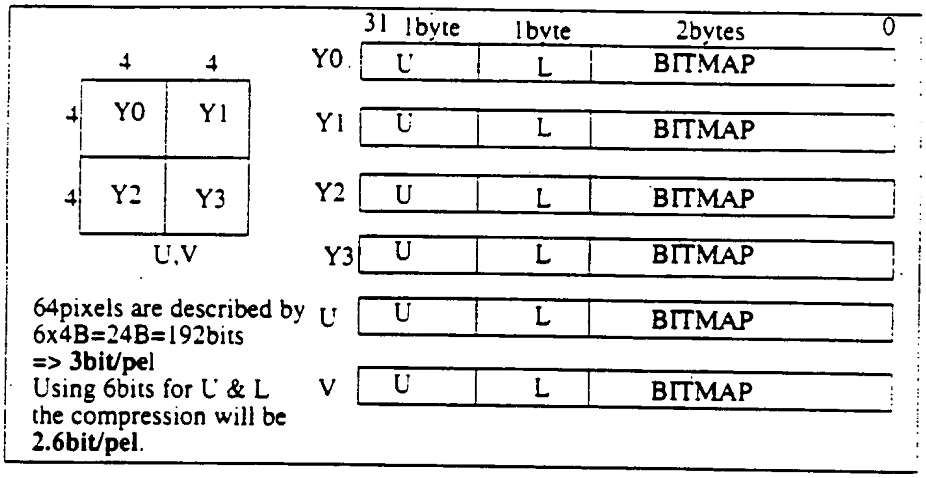
These severe memory bandwidth limitations lead us to attempt to minimize video memory traffic. If we store **Cinepak data in 4:2:2, with an avarage of 16bit/pel**, independent of Cinepak window size, **Nordic-1M will not be able to run Cinepak above 800x600.** Any attempt to execute vertical zoom with interpolation, even with two points interpolation will further aggravate this memory band-width bottleneck. The next step should be 100MHz SDRAM-s with a large CRT FIFO (32x32) or 64bit wide memory data bus. These considerations apply to any playback compression standard that stores data in video-memory as 16bit/pel in avarage. The key to a successful compression standard would be one that does decompresssion on the CRT memory read and stores data in memory as 8bit/pel or less, in avarage.

As we store live video at 2.6bit/pel and fetch it as if it was 8bit/pel with Sashapak, Nordic-1M should be able to run live video even at 1024x768.

If we defined a new CDROM play-back standard based on Sashapak, then we could run even 1024x768 with 24bit/pel accuracy and keep it memory as an 8bit/pel. **Keith and Ken are actually working on it, with Sasha's help.**

4.4.1 On Sashapak

Sashapak is a highly assymetric, lossy "Block Truncation" compression algorithm optimized for visually equivalent image processing.
Data is proccessed as Y,U,V one byte each. U and V are sub-sampled, characterizing one 8 by 8 pixel array, while Y is characterizing a 4x4 pixel aray. Actually one of two values is assigned to each pixel inside the 4x4 (for Y) and 8x8 (for U and V).
At compression time, the compression chip determines a threshold and two values U&L for each group of sixteen (4x4) pixels. This way all pixels above threshold will be assigned the value U, all pixels below threshold will be assigned the value L.



At decompression, the bitmap will control a mux between the U and L values. This way the decompression is straightforward.
The main drawback of this approach is that a BITMAP contains Y data from 4 scan lines and U,V data from 8 scan lines. When data is read from memory to be displayed only data from one scan line is used. Similarly the U/L values apply to several scan-lines and the same value will be read in each scan line. The same data will be read again and again in each scan line raising the actual memory bandwidth requirements to an equivalent of 16bit/pel map.
In order to overcome this problem and reduce memory bandwidth requirements, Sasha proposed a **scan line oriented, segregated mode of data organization**. This would require the Sashapak compression chip to write compressed data in a specific way puting the strain on the compresion chip address generator. The basic principles of the new organizations are:
1. Data is organised in the MV Window in chunks of 32 sequential pixels.
a)For **8bit U/L resolution**, as one Y covers 4 pixels, 8 U/L sequential values are needed, 16bits each -> **8x16/32=4W** (8x12/32=3W for 6bit res. U/L) (where W is a 32bit word).
b)U and V U/L data is packed together in the same word (16bits for U U/L and 16 bits for V U/L -> total 32bits=1W) and as one pair of U and V applies to 8pixels, 4 such values are needed for 32 sequential pixels -> **4x32=4W**. (4x12x2/32=3W for 6bit res. U/L)
c) The mask data is also scan-line oriented: all 32bits of Y mask of sequential 32pixels are placed in one 32bit word and due to sub-sampling on U and V, all 32bits of U and V mask of sequential 32pixels are placed in one 32bit word. So in 2W data for 32 sequential pixels on one scan line is packed. -> **2W** (the same for 6bit resolution for U/L)

d) For 640x480 window size, even at 8bit U/L resolution, data for 8 scan-lines fits in one DRAM page (symetric DRAM-s assumed): 320WY+80WU+80WV=480W < 512W page

← 480W for one 8x640 pixels block fits in one page →					
SLA=MVW START-n*MVW-Offset	U/L for Y	U/L for U,V	Y bitmaps	UV bitmaps	8Bit per U/L resolution.
	160W SLA+A0h	80W SLA+F0h	160W SLA+190h	80W	Data Grouped per Scan Lines
Where Si is scan-line number i inside the block			S0 S1 S2 ...	S7 S0.1 .... S6.7	

f) The key to memory bandwidth optimization is the fact the the compression chip will organize data sequentially over 32 pixels instead of 8. This will be done for the mask bits which will be organized sequentially by scan line and for the U/L data. This way a 32bit word of Mask data contains only information for the current scan line and no extra memory fetches are required. The main reason we segregate U/L for Y and U/V is the fact that we want to use 6bit U/L. In this case it is easier to identify the proper U/L info chained within 32bit words if the words are sequential and we have a fix relationship between their address and the 6bit U/L quantities.

g) If we calculate now the amount of data we need to fetch to display 32 sequential pixels:
- for 8bit per U/L: 4W (YU/L) + 4W (UV U/L) + 1W (YM) + 1W (UVM)=10W=40B
  --> 40B/32pixels=320bit/32pixels=10bit/pel to fetch from memory
- for 6bit per U/L: 3W (YU/L) + 3W (UV U/L) + 1W (YM) + 1W (UVM)= 8W=32B
  --> 32B/32pixels = 8bit/pixel

This is a very good data fetch ratio, much better than the 16bit/pel we'd get without the optimization.

g) When reading data from the Motion Video Window the controller will generate the following sequence of addresses, all of them in the same DRAM page (8bit per U/L):
- fetch the YU/L for the first 32 pixels on scan-line n at addresses:
  MVW_Start+n*MVW_offset, +1, +2, +3
- fetch the UV U/L for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+A0h, +1, +2, +3
- fetch the Y BitMap for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+F0h
- fetch the UV BitMap for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+190h
- fetch the YU/L for the first 32 pixels on scan-line n at addresses:
  MVW_Start+n*MVW_offset+4, +5, +6, +7
- fetch the UV U/L for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+A0h+4, +5, +6, +7
- fetch the Y BitMap for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+F0h+1
- fetch the UV BitMap for the first 32 pixels on scan-line n at address:
  MVW_Start+n*MVW_offset+190h+1
- now the MVW FIFO is full.
- chunks of ten 32bit words will be proccessed for every 32pixels displayed
- the MVW FIFO gets empty after the first 10 words being read for decompression and display.

Please note that all data for every 8 scanlines starting from the top of the MVW is in the

same DRAM page.
Because we have to fetch the MVW Data before we start displaying it, and because at that time the CRT FIFO does not empty as fast as it is needed (unless MS Windows is running at the same pixel depth as the MV Window), it is necessary to have a separate FIFO for the MVW or at least half the FIFO (at least 10W for 8bit per U/L and at least 8W for 6bits per U/L).
h) When we run 6bit per U/L, because the Video Memory band-width requirements are as if we were running 8bit/pel, we could do vertical zoom with avaraging. But there should not be a need for zoom with Live Video because there is enough CPU and memory band-width to run 640x480 windows even when MS Windows is run as 1024x768 4 or 8bit/pel.

Conclusion: With Sashapak we can get data organized in memory such that the memory bandwidth is as for 10bit/pel (8bit U/L) or 8bit/pel (6bit U/L) pixel depth. The actual memory area will be close to 3bit/pel or 2.6bit/pel depending on the resolution of U/L values - 8 or 6bits per value.
Please note that the memory area will be slightly larger due to the fact that we use only 480 words in a page of 512.

4.4.2 YUV to RGB Conversion Algorithm

The ideal YUV to RGB transformation is:
R=Y+1.37V
B=Y+1.73U
G=Y-0.699V-0.336U

After defining an objective way of measuring color error, Sasha came up with the following Conversion algorithm:
R=Y+1.375V=Y+1 3/8 V
B=Y+1.75U=Y+1 3/4U
G=Y-0.375U-0.75V= Y-(3/8U+3/4V).

This algorithm can be implemented with 8 9bit adders and it should support excess 128 or two's complement U,V formats (U,V may be negative while Y is always positive).
This algorithm will be used with Live Video.
For Cinepak, the YUV->RGB converter will be reconfigured to the Cinepak YUV->RGB conversion rule:
R=V+Y
B=U+Y
G=Y-V/2-U/4

Cirrus Confidential Business Information

CL 152700

4.4.2 Motion Video Architecture Functional Blocks

Instead of treating play-back and live-video as two separate modules, we'll isolate several generic functions to be used by both:

- Display Window Generation Block:

allows s/w to define one or two play-back or live video **window(s)**.

The MV Window is defined in memory cycles relative to the start of line in horizontal and frame in vertical direction. Horizontally the unit of measure is different outside relative to the inside of the window: the horizontal window start coordinate is expressed in memory cycles for the surrounding pixel depth (MS Windows pixel depth, normally 4 or 8 bit/pel), while the width of the MVW is expressed in memory cycles for the MVW pixel depth (8bit RGB, 16bitRGB, 422YUV spacial, 422 YUV linear, 411YUV linear, Sashapak).

- **XS** = MVW Horizontal Start Coordinate (in surrounding pel depth memory cycles)
  The horizontal position is programmed with 8 pixel resolution (0,8,16,... pixels starting from the left side of the screen)
  XS register CR34[7:0] -> 8 bits
  (up to 1K pels in increments of 8 pels)
  To get an early warning, we will require XS to be programmed 8 pixels less than the actual position: 0 is zero, 1 is 16pels, 2 is 24 pels,...
- **XW** = MVW Horizontal Width (MVW pel depth memory cycles)
  XW register CR35 [7:0] -> 8 bits
  (up to 1K pels in increments of 8 pels)
  To get an early warning, we will require XW to be programmed 8 pixels less than the actual width: 0 is zero, 1 is 16pels, 2 is 24 pels,... width.

- **YS** = Vertical MVW Start (in true scan-lines not affected by scan-line doubling or panel vertical expansion)
  10 bits in CR36[1:0] (msb-s) and CR37[7:0] (lsb-s).
- **YE** = Vertical MVW End
  (all bits are programmed, in true scan lines)
  10 bits in CR36[3:2] (msb-s) and CR38[7:0] (lsb-s).
- **SAdOf=** Surrounding Address Offset (a number to be added every line to the surrounding CRT Address Counter value to allow it to jump over the MVW without counting through it. This allows calculating the surounding restart address.) This number depends on the surrounding mode resolution. Assuming a maximum line of 1024 pels, in increments of 4 pels per address (8bits/pel), 256 addresses is enough -> 8bits.
  8bits in CR39[7:0]
- **WMAdS** = MVW Memory Address Start in increments of 16KB = 4KW (a phisical address 00000:3FFFF for 1MB or 00000:7FFFF for 2MB Video Memory).
  7bits in CR3A[6:0]

- **WMAdOf = MVW Memory Address Offset** (A number to be added to the current MVW Start Address to get the next MVW start Address). Having this s/w model allows panning through a large image for the MVW. The actual image stored in memory may be as large as 1K pixels

at 16bit per pixel (2 per word) -> 9bits.
9bits -> in CR36[4] (msb) and CR3B[7:0] (lsb).
Because the CRT Address Counter counts surrounding memory cycles, during the MVW display it would count a wrong number corresponding to the MVW pixel depth. To avoid this wrong count, the CRT Address Counter is stopped while MVW is displayed and loaded with the value corresponding to the end of MVW and restart of the surrounding display. As this address value changes every line, an offset is specified and will ba programmed by the MS Windows driver depending on the MVW size and pixel format. The horizontal size of the window is controlled by a different counter counting to XW. During the MVW display the adddress will be generated by the same CRT Address Counter which will be loaded with a MVW Memory Address Start (WMAdS).

- **XW8P = MVW Horizontal Width in 8 pixel units**. Defines the size of the window in 8 pixel units. It is used to terminate the window horizontally. It may require to program the value minus one or two. To be Defined Later. Register CR3D[6:0] will contain this value.

Every scan-line

MVW Scan-Line Start Address = Previous Scan-Line MVW Start Address + WMAdOf

Scan-line doubling for "zoom" should be taken into account when designing the MVW Architecture: on the address generation block.

- **Off-screen MV Memory Addressing and Access Control.** This includes data type tag generation based on MV memory Address and the circuitry placing the tags in the CRT-FIFO. This block has provisions for scan line replication by repeating the MV address generated on the previous scan line of the MV window (only).

As Cinepak needs 4:1:1 rectangular format the MV Memory Address Counter needs to count in a non sequential way:

Y0Sn Y1Sn Y2S(n+1) Y3S(n+1) U V

implies an address counter that can skip over two numbers, or maybe **a second address counter that starts counting from 2 instead of zero.**

- **One tag bit is generated based on CRT Address and is 1 if the data in the CRT FIFO is from the MVW, other-wise is 0.** This tag bit, called the **steering bit** will be delayed through the entire data path and end up controlling the final video data mux just before the DAC.

Other tag bits called "data-type" tag bits encode the type of 32bit word in a given data format and are used by MVW Decoder/Serialiser to steer CRT-FIFO data at CRT-FIFO read.

For instance, in 4:1:1 linear data format, data is stored in memory and in the CRT-FIFO as Y0Y1Y2Y3 U03V03Y4Y5 Y6Y7 U47V47. 3 groups of 32bits. Two "data-type tag bits will mark the three types of 32bit words so that steering logic knows where to place the three words in the data serialiser.

Similarly Sashapak will need 3 "data-type" tag bits to encode 8 32bit words (for 6bit per U/L) or 4 "data-type" tag bits to encode 10 32 bit words (for 8 bits per U/L).

16bit RGB 5:6:5 and 5:5:5 and 8bit RGB 3:3:2 can either use the **standard VGA data**

path. which in 5428 was extended to 16bit wide (used for 16bit/pel 1kx768 interlaced) or the new data path. **if a phisically new data path will be actually created**.

- **MV Data Path** . This is area wise the largest piece. It consists of a new 16bit wide data-path with the same delay as the VGA data path. that takes MV data from the MV prefetch cache and the CRT FIFO. treats it appropriately depending on the MV data format encoded in the MV tags and converts it to 24 bit YUV first and 24bit RGB second. Horizontal zoom with avaraging (or maybe 5 levels of interpolation) is in this block too.

- **Larger (26 or 24 stages) CRT-FIFO, Variable CRT-FIFO threshold and its control**. Because the MS Windows pixel depth and MVW pixel depth don't match. it is necessary to reserve CRT-FIFO space to fill in enough high pixel depth data before the MVW display starts. In Sashapak this means 10 or 8W (for 8 or 6bits per U/L respectively).

This function can be implemented as a dinamic threshold size modification of the CRT-FIFO. If the CRT-FIFO has 24 stages instead of 16 but only 16 stages are used before the MVW but it is switched to 24 when MVW fetches start, there will be 8 more stages to fill at that time ensuring more prefetched data in the CRT-FIFO at the beginning of each MVW scan-line.

In normal operation the threshold is 8 (or 6).

Here is the sequence of events that follow the detection of MVW start:

1. Drop CRT-FIFO threshold to 1 to get immediat service & make 10 (or 8) more stages available in the CRT-FIFO.

2. Start fetching MVW data till CRT-FIFO is full. (BLT in progress will increase latency)

3. Increase threshold to 10.

4. Proceed as such till end MVW.

5. Load the CRT Address Counter with its last contents + SAdOf

(where SAdOf = Surrounding Address Offset)

7. Continue fetching based on the new address (start with random cycle). CRT-FIFO threshold remains the same - at least 10 or 8. CRT-FIFO size is still 26 or 24.

8. At the end of line, upon flushing the CRT-FIFO before prefetch, decrease CRT-FIFO size back to 16 keeping the threshold the same.

9. Go back to #1 for the next scan-line.

This sequence is applied if MVW pixel depth is higher than surrounding pixel depth. If MVW pixel depth is lower than surrounding pixel depth, then the action described at #1 above takes place at the end of MVW, instead of the beginning. ***In other words, FIFO depth extension takes place when a switch from low to high pixel depth occurs.*** **If MVW pixel depth is the same as the surrounding, only the address is switched. In this case the CRT FIFO depth and the threshold remain the same.**

- Steering logic decode the tags comming out of the special addition to the CRT-FIFO and the pre-fetch buffer and controls the decompression. formatting and serialization.

(they tell the formatter what data is Y, U or V).

- What about the video serializer load and the CRT FIFO read signal? What hapens to them when in MVW? They are one signal today.
WHEN the data path switched to display MVW data, what hapens with the VGA Video Data Path is don't care as we do not display that data, except for the h/w cursor anf the h/w icon that do not go through the standard VGA serializer. So we'll probably stop the VGA data path untill the point h/w cursor and h/w icon are combined into it, to save power, as we will stop the MVW data path when not in use, for the same reason.
So, the Video Serializer Load may be stopped while MVW data is displayed, **but the CRT FIFO read will be going on, with a frequency and shape depending on the data format loaded in the CRT FIFO for display.**
In RGB 3:3:2 8bit/pel direct color mode CRT-FIFO read will be generated every 4 vclks.
In standard RGB 5:5:5 or 5:6:5 16bit/pel modes CRT-FIFO read will be generated every two vclks.
In YUV 4:2:2, 24bit/pel (Y0 U Y1 V on one scan line) the CRT-FIFO read will be generated also every two vclks.
IN Sashapak 10 (for 8bit per U/L) or 8 (for 6bit per U/L) CRT-FIFO read pulses will be generated every 32 vclks.
- The MVW Address Counter, loaded at the beginning of each MVW scan-line with a start address calculated based on the Start MVW Address and the MVW Address Offset, will count at format dependent rate for as long as the CRT-FIFO is not full. Because the CRT-FIFO will be emptied faster in the MVW (in general the pixel depth will be higher in the MVW) there will be requests for more memory cycles when displaying in the MVW.

- If either h/w cursor or h/w icon are on, they should be displayed even if the direct data path is used. In this case both video data pathes will clock and the steer tag will point to the VGA data path when either h/w cursor or h/w icon or both are displayed, even on top of a MV window. Please note that neither the h/w cursor, nor the h/w icon data go through the CRT FIFO or the VGA Data path except for the final pixel address block and the RAMDAC RAM, though they have a special path in the RAM. So we could stop most of VGA data path (including the internal palette) but not all of it even if we are in 16bit/pel mode (the new one that goes through a separate 16bit data path).

- If we are in a 16bit/pel mode with no h/w cursor or h/w icon, or in a 640x480 MVW mode filling the screen, we can stop the VGA data path to save power.

- The off-screen MVW memory will be a programable start memory array normally placed towards the end of the memory, but before h/w cursor, h/w icon and the half frame buffer, even the color one. The start of the MVW memory will be programable in increments of 4KW (16KB) as a physical address -> 7bits to program.

CR3B[6:0] is the MVW Memory Array Start Address.

2MB configuration will support enough memory to support:

- one 640x480 3bit/pel window or up to two 320x240 3bit/pel (Sashapak) -> 32KW
- one 640x480 16bit/pel window or two 320x240 16bit/pel windows

(153.6KW=614.4kB out of 256KW or 512KW available depending on the memory con-

figuration).

- all of the above at the same time up to a total of two windows though.

1MB configuration will support:

- one 640x480 3bit/pel Sashapak window or two 320x240 Sashapak windows or
- two 320x240 16bit/pel windows.

Please note that about 24KW are needed for 800x600 DS Color STN HFB, the h/w cursors and h/w icons while about 18KW are needed for 640x480 DS Color STN HFB and that

- 640x480 4bit/pel requires 38.4KW
- 800x600 4bit/pel requires 60KW
- 1024x768 4bit/pel requires 98.3KW
- 640x480 8bit/pel requires 76.8KW
- 800x600 8bit/pel requires 120KW
- 1024x768 8bit/pel requires 196.608KW

Video Motion Formats supported are:

- 8bit RGB 3:3:2 going through the palette (RGB 3:3:2)
- 16bit RGB (5:5:5 Sierra and 5:6:5 XGA) not going through the palette -> this is a 16bit/pel that does not require double dotclock !!
- 4:1:1 array converted to 4:2:2 Y0 Y1 Y2 Y3 U V -> Y0UY1V Y2UY3V for Cinepak (with Y2Y3 for the next scan line)
- 4:2:2 linear Y0UY1V from TV Decoder
- Sashapak format (we'll support only 6bit/pel)
  - for 8bit per U/L:
    4W (YU/L) + 4W (UV U/L) + 1W (YM) + 1W (UVM)=10W=40B
    --> 40B/32pixels=320bit/32pixels=10bit/pel to fetch from memory
  - for 6bit per U/L:
    3W (YU/L) + 3W (UV U/L) + 1W (YM) + 1W (UVM)= 8W=32B
    --> 32B/32pixels = 8bit/pixel

CR3C[3:0] will encode the pixel format for the first motion video window data:

0h = 0000 = 8bit RGB 3:3:2 going through the palette
1h = 0001 = 16bit RGB Sierra (5:5:5)
2h = 0010 = 16bit RGB XGA (5:6:5)
3h = 0011 = 4:2:2 YUV Y0UY1V
4h = 0100 = 4:1:1 YUV linear Y0Y1Y2Y3UV
5h = 0101 = 4:1:1 YUV array Y0Y1Y2Y3UV but Y2Y3 are in next scan line
6h = 0101 = Sashapak 6bits for U/L
7h = 0110 = Sashapak 8bits for U/L
8h:Fh = 0111:1111 = reserved

Cirrus Confidential Business Information

CR3C[4] = Enable Motion Video Window
CR3C[5] = Enable Live Video in Full Screen
CR3C[7] = MVW horizontal zoom (doubling)
CR3C[6] = MVW vertical zoom (doubling)

CL 152705

Note: We will not support 4:1:1 linear from TV decoder, for design simplicity.

Memory data formats:

- 8bit and 16bit RGB, as in 5428.
- 4:1:1 Cinepak converted to 4:2:2 by writing a 4:2:2 Code-Book

first scan line -> YOUY1V -> One 32bit Word

---

second scan line in memory -> Y2UY3V -> One 32bit Word

- 4:2:2 linear will be mapped in memory as follows:

first scan line YOUY1V
first scan line Y2UY3V
---
consecutive data in the same scan line.

The tags that accompany the motion video data through the CRT-FIFO mark both the data format and the type of data inside the format (U/L for Y or U,V or mask for Y or U,V in Sashapak, Y0, Y1,U or V in 4:2:2).

Double Buffering for MVW Video Memory Data

In order to get a high quality Live Video full frames should be displayed at all times. This assumes that writing the video buffer and displaying it are in sync, which is normally not the case..

To solve this problem and provide a simple interface between MVA h/w and the driver. Nordic will support a double buffer approach.

Addressing with Sashapack:

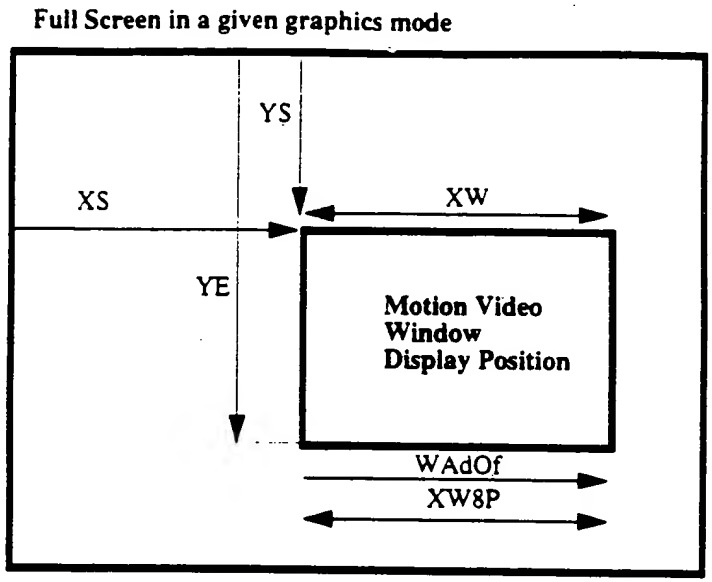
Sashapack stores data for eight scan-lines in one memory page (works only with synchronous DRAM-s). But due to U,V sub-sampling diferent data types are not read from memory uniformly.

On Y U/L the first four scan-lines read at offset 0 in the page, but the second four scan-lines read at offset 50H in the page.

On U,V U/L we read the same data for each scan-line. So we repeat reading the same data at all times, for eight scan-lines. Page addresses to read are A0:EF. If the scan-line is shorter it will end sooner.

On Y Map the data should be sequential, scan-line oriented: first pixels 0 to 639 on the first scan line, than pixels 0 to 639 on the second scan line and so on till the 8-th scan line. If the lines have less than 640 pixels, Sasha wants them cropped for the first 4 scan lines, and then starting again from the half for the next four scan-lines. This is in sync with Y U/L where the last four scan-lines start from mid.

On U,V Map, because of the vertical subsampling, the same data is read for two scan lines, so there are four scan-lines worth of data but each scan-line is read for two consecutive scan-lines. Even if the scan-line is shorter, there will be a gap every-scan-line: there are fix page addresses for each scan-line up to 640 pixels wide.



Where:

- **XS** = MVW Horizontal Start Coordinate (in surrounding pel depth memory cycles)
- **XW** = MVW Horizontal Width (MVW pel depth memory cycles)
- **YS** = Vertical MVW Start (in true scan-lines not affected by scan-line doubling or panel vertical expansion)
- **YE** = Vertical MVW End (all bits are programmed, in true scan lines)
- **WAdOf** = MVW Address Offset (a number to be added every line to the CRT Address Counter to allow it to jump over the MVW without counting through it.
- XW8P = MVW horizontal width expressed in 8 pixel units

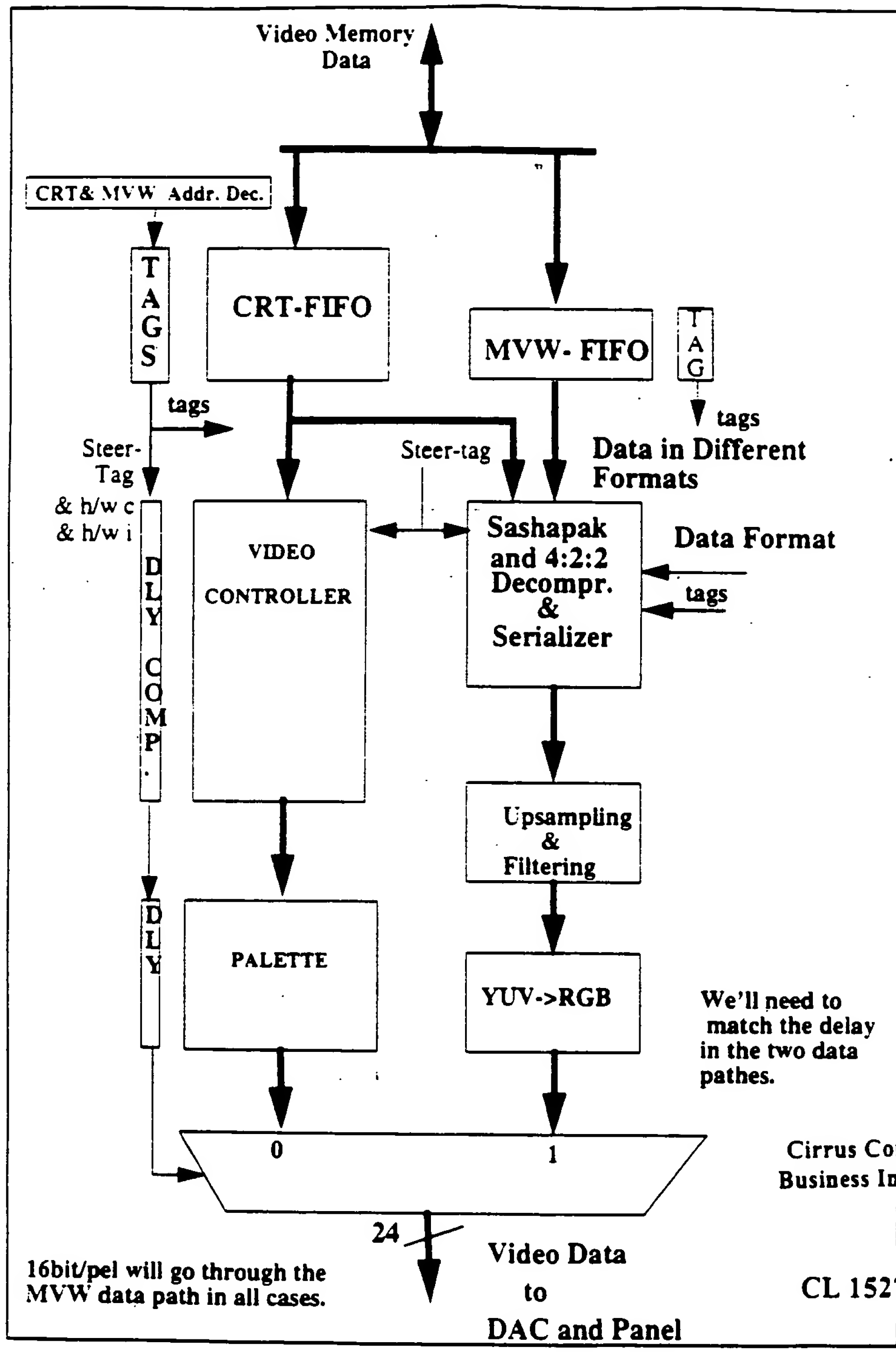
Please note that MVW horizontal width is defined twice in different units:

XW defines it on the memory side in memory cycles

XW8P defines it on the CRT side in 8 pixel units.

Cirrus Confidential Business Information

CL 152707


Video Memory Data
CRT& MVW Addr. Dec.
TAGS
CRT-FIFO
MVW- FIFO
TAG
tags
tags
Steer-Tag
& h/w c
& h/w i
Steer-tag
Data in Different Formats
DLY COMP.
VIDEO CONTROLLER
Sashapak and 4:2:2 Decompr. & Serializer
Data Format
tags
Upsampling & Filtering
DLY
PALETTE
YUV->RGB
We'll need to match the delay in the two data pathes.
0
1
24
16bit/pel will go through the MVW data path in all cases.
Video Data to DAC and Panel

4.5 Feature Connector Support

Nordic-1M will support 5428 Feature Connector so that it is compatible to Media Manager.
Nordic-1M will support VAFC (Vesa Advanced Feature Connector) in the 8bit VSVPC passthrough compatibility mode.
The main difference beteen 5428 FC and Nordic FC is the fact that 5428 FC has one EDCLKn control pin controlling the direction of the DCLK pin, while Nordic-1M as well as VAFC have a FCDCLK output pin and a FCVCLK input pin.
Dave Keene believes that 5428 FC is VAFC compatible. To ease the design. I kept 5428 FC pin names (not VAFC pin names).
On VAFC compatibility here are some comments:
- GENCLK can be applied on XVCLK if SR22[3]=1.
But, if SR22[3]=1 (select xclk) & SR24[7]=1 (VAFC enabled), then XMCLK should get OSC and MCLK Synthesizer operates normally, except that the reference frequency comes from XMCLK instead of OSC pin. If SR22[3]=1 & SR24[7]=0, then MCLK comes directly from XMCLK pin and the MCLK and VCLK Synthesizers are powered-down !!
- GRDY signal is similar to OVRW.
- VRDY is similar to EVIDEO if EVIDEO is dinamically switched as a valid data in signal.
- EGENn is not needed if we configure Nordic-1M in the right configuration with SR22[3] (XCLKPU) and SR24[7] (VAFCPU).
- As VAFC maximum FCVCLK frequency is 37.5MHz, VAFC supports a mode in which data is sent at 37.5MHz to the DAC and it each pixel will be displayed twice if the chip is running at 75MHz pixel clock. This allows only 8bit/pel data to be displayed. To display 16bit/pel data we should use both phases of the FCVCLK and pack the data into one pixel. I think that this is already done in 5428, if not Man-Shek and Fong-Jim are modifiying the 5429 RAMDAC to support this mode -> we need to support it too.
- VAFC requires FCDCLK to be switchable under s/w control beteen 1x and 1/2x the pixel clock. If we took this out of Nordic-1M data base, we should put it back, but the divide by two should affect only FCDCLK going out of the chip.

Check VESA VAFC Proposal 1.0p rev. 0.31 07/15/93 for VAFC timing info.

Nordic-1M FC Pins Functional Description

Most of this is from 542X Data Sheet page 3-26:

Cirrus Confidential Business Information

VSYNC and HSYNC TO The standard CRT pins tri-stated when FCESYNCn is L. Thes pins have programable polarity.

FCBLANKn I/O BLANKn in 5428. If FCESYNCn is H. this is an output and it supplies the actual composite BLANKn CRTC signal. VAFC requires this to be the composite Display Enable (no borders). If this is the case, we'll need a bit for VAFC that puts DE = HDE & VDE inverted on BLANKn.

CL 152709

If FCESYNCn is L, FCBLANKn is an input and it forces RAMDAC RAM outputs to 0 (black), but it should force them black before they go to the panel.
It seems that by connecting OVRW output to BLANKn as input we can display FC data in a window - this will work though only if OVRW, internally controls the pixel address mux between normal VGA data and the FC data.

FCP[7:0] I/O P[7:0] in 5428.
If FCEVIDEOn is H, these pins are outputs and represent the pixel address of the RAMDAC.
IF FCEVIDEOn is L, these pins are inputs and represent the pixel address to the RAMDAC.
CR1A[3:2] control the way the FCP[7:0] are muxed with the normal data.

FCDCLK O (the actual pin is I/O) outputs VCLK or VCLK/2 if selected through s/w.

FCESYNCn I ESYNCn in 5428. Controls HSYNC, VSYNC and FCBLANKn (H-> outputs, L-> H/VSYNC are tri-stated, FCBLANKn is tri-stated.).

Nordic does not have EDCLKn input as FCDCLK and FCVCLK are not shared.

OVRW O OVRW in 5428 = Overlay Window. **This active Low** signal, is generated by the BLANKn VT and HT logic (shared) and is supposed to be used to create an Overlay Window in which data from the Feature Connector is displayed. *I don't know why we output this signal. It should be controlling the MUX between the normal pixel address and the FC originated pixel address.*

4.5.1 On the Video Port for TFT Panels

A Video Overlay live video solution requires a TV Decoder Chip Set, a de-interlacer (at least a line buffer, but usually a full frame buffer), a scalar/filter/color converter chip (like CL-PX0072, CL-PX0070 or the more expensive CL-PX2070). To display live video in a window at 800x600 or 1024x768 a full frame buffer is needed and some frame level synchronization needs to be achieved.

Is there a cheap way to display full screen TV on a 640x480 panel?
Is a video port a cheap system solution to the above problem?

Can we design a system that uses a universal TV Decoder like SAA7151A and feeds data directly to a Video Port that displays live video at 640x480 30fps with no additional components? This is what customers require!

Problems to solve:

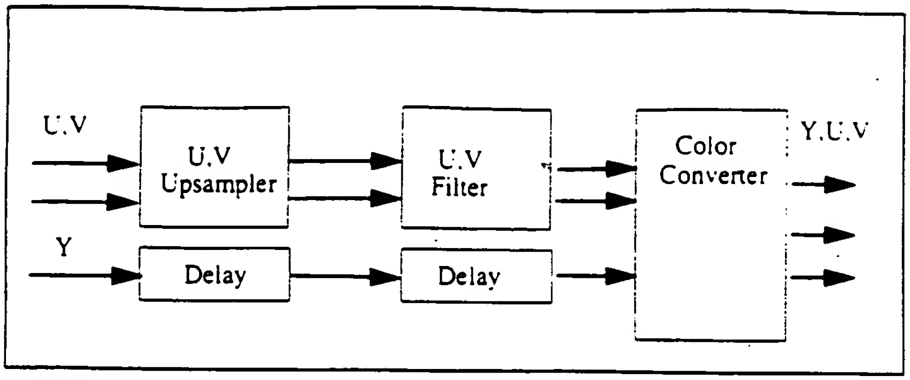
1. Accept 7151 data format: 4:2:2 YUV at 6.75MHz or 4:1:1 YUV at 3.375MHz, interlaced.
2. Synchronize at frame level.
3. De-interlace. Optionally display only odd fields with scan line doubling (according to Sasha this looks better than displaying true de-interlaced picture).
4. Get enough memory bandwidth with a 32bit wide DRAM.
5. Have enough storage space on the Graphics Controller to go over the CRT horizontal non-display memory refresh & prefetch time while Live Video Data comes at normal TV display speed.

4.5.1.1 Data Format

In order to accept YUV data, we need good up-sampling filters on U and V. After up-sampling and filtering we need a Color Space Converter with 2-th complement (for 7151) or excess 128 (for other TV decoders). A good 4:1:1 UV up-sampling filter is quite complex and requires many stages of pipe-lining (9 or 11). A good 4:2:2 UV up-sampling filter requires less than 5 stages of pipe-lining.

As data is stored in TV decoder format, up-sampling, filtering and color conversion are done on the CRT read side at dot-clock rate.

Please note that 4:1:1 data requires 48 bits of storage (two data packs will fill 3 32 bit words).



4.5.1.2 Synchronize at Frame Level

To avoid moving objects jagging we should display only full frames. In many systems today this requirement is not met. For instance play-back systems display as fast as possible without any regard for frame synchronization. According to Sasha this is not good for TV. How bad it is remains to be seen.
If the CRT/TFT display runs at 27MHz (double TV dot-clock rate) it is possible to use a double buffer scheme in which we display twice each frame while writing one TV frame in the other video buffer and alternate the two buffers. For a 640x480 display at 4:2:2 YUV we'd need 2MB of DRAM, as the buffers are embedded in the Video Memory (614.4KB per buffer needed).
Even with this scheme, external VSYNC from the TV Encoder should be used to synchronize (= reset) Nordic vertical counter, so that TV and Graphics frames start at the same time. SAA7151 outputs VS (TV VSYNC) which can be used as external SYNC by Nordic.
If only odd fields are displayed, half data is stored reducing Video Memory requirements to 307KB per frame (614KB total which fits in 1MB). This is not a bad idea especially if the picture looks better!

4.5.1.3 De-interlace

This is naturally done by writing data in the Video Memory. If only odd fields are displayed, Nordic has to determine the field boundary by counting HS (TV HSYNC) pulses.

4.5.1.4 Memory Bandwidth with 32bit Wide DRAM

There are two cases: 4:2:2 YUV and 4:1:1 YUV.

A. 4:2:2 YUV Bandwidth Requirements

CRT-dot-clock frequency = 27MHz. TV-clock frequency = 13.5MHz
R=6m, P=2m

a. TV-FIFO = 16 stages with 8 to write 8 to read .

CRT-FIFO = 16 stages with 8 to write and 8 to read
2xCRT-FIFO-fills + TV-FIFO-empty < 2x16 CRT-dotclks=16 TV-dotclks=1185ns
2x(R+7P) + R+7P + 3m = 3R+21P+3m=18m+42m+3m = 63m < 1185ns
mclk-period < 1185ns/63 = 18.8ns -> 53MHz min mclk frequency

b. TV-FIFO = 32 stages with 16 to write and 16 to read.
CRT-FIFO = 16 stages with 8 to write and 8 to read
4xCRT-FIFO-fills + TV-FIFO-empty < 4x16 CRT-dotclks=32 TV-dotclks=2370ns
4x(R+7P) + R+15P + 5m = 5R+43P+5m=30m+86m+5m = 121m < 2370ns
mclk-period < 2370ns/121 = 19.58ns -> 51MHz min mclk frequency

c. CRT-FIFO = 32 stages 16 to write and 16 to read @ 27MHz
TV-FIFO = 16 stages 8 to write and 8 to read @ 13.5MHz

CRT-FIFO-fills + TV-FIFO-empty < 32 CRT-dotclks=16 TV-dotclks=1185ns
(R+15P) + R+7P + 2m = 2R+22P+2m=12m+44m+2m = 58m < 1185ns
mclk-period < 1185ns/58 = 20.43ns -> **48.9MHz min mclk frequency**

Please note that Nordic already has 32 (or 36 stages of CRT FIFO - MVA FIFO) and 24 stages of FB-FIFO (write + read). So if we use all the buffering area available we can meet case c requirements and run direct live video at 48.9MHz mclk.

d. CRT-FIFO = 24 stages (12/12)
TV-FIFO = 12 stages (6/6)
CRT-FIFO-fills + TV-FIFO-empty < 24 CRT-dotclks=12 TV-dotclks=888ns
(R+11P) + R+5P + 2m = 2R+16P+2m=12m+32m+2m = 46m < 888ns
mclk-period < 888ns/46 = 19.3ns -> 51.8MHz min mclk frequency

B. 4:1:1 YUV Bandwidth Requirements

a. CRT-FIFO = 24 stages (12/12)
TV-FIFO = 12 stages (6/6)
CRT-FIFO-fills + TV-FIFO-empty < 32 CRT-dotclks=16 TV-dotclks=1185ns
(R+11P) + R+5P + 2m = 2R+16P+2m=12m+32m+2m = 46m < 1185ns
mclk-period < 1185ns/46 = 25.7ns -> 38.8MHz min mclk frequency

Because 3x32bit house 8 pixel data, it is advisable that the number of stages in the FIFO-s are a multiple of 3 for 4:1:1.

For 4:1:1 YUV, the picture quality is not that good as for 4:2:2, the control requirements are harder to meet, the filter on U and V is expensive, but the bandwidth requirements are easy to meet.

4.5.1.5 Covering the Memory Refresh and CRT-FIFO Prefetch Time

During each horizontal non-display time, Nordic executes memory refresh (normally 3 with an option of 1 - all random cycles), h/w icon and CRT-FIFO prefetch cycles.

At that time the TV data comes at normal rate and has to be stored in the TV-FIFO. Let's see how long it takes to execute all these cycles.

a. With reference to case A.c. 4.5.1.4
(4:2:2 YUV the case that meets bandwidth requirements at 48.9MHz mclk)

x. CRT-FIFO = 32 stages, 3 refreshes, h/w icon 64x64x2 (4 h/w icon cycles)
R+31P + 3R + R+3P + 3m = 5R+34P+3m=30m+68m+3m=101m < 1185ns
-> m < 1185ns/100m=11.7ns -> min mclk freq. is 85MHz
(off by about 42mclks to get to 50MHz mclk).

y. CRT-FIFO = 32 stages, 3 refreshes, h/w icon but prefetch 16 CRT-FIFO stages
(no need to fill the FIFO at the beginning of the line).
R+15P + 3R + R+3P + 3m = 5R+18P+3m=30m+36m+3m=69m < 1185ns
-> m < 1185ns/69m=17.17ns -> min mclk freq. is 58.2MHz

z. CRT-FIFO = 32 stages, 1 refresh, h/w icon but prefetch 16 CRT-FIFO stages
(no need to fill the FIFO at the beginning of the line & use extended refresh DRAMs).
R+15P + 1R + R+3P + 3m = 3R+18P+3m=18m+36m+3m=57m < 1185ns
-> m < 1185ns/57m=20.78ns -> **min mclk freq. is 48.1MHz**

This case meets the minimum requirements for non-display time buffering.

b. With reference to case B.a. 4.5.1.4
(4:1:1 YUV the case that meets bandwidth requirements at 38.8MHz mclk)

x. CRT-FIFO = 24 stages, 3 refreshes, h/w icon 64x64x2 (4 h/w icon cycles)
R+23P + 3R + R+3P + 3m = 5R+26P+3m=30m+52m+3m=85m < 1185ns
-> m < 1185ns/85m=13.9ns -> min mclk freq. is 71MHz
(off by about 26mclks to get to 50MHz mclk).

y. CRT-FIFO = 24 stages, 1 refreshes, h/w icon but prefetch 16 CRT-FIFO stages
(no need to fill the FIFO at the beginning of the line & extended refresh DRAMs).

R+11P + 1R + R+3P + 3m = 3R+14P+3m=18m+28m+3m=49m < 1185ns
-> m < 1185ns/49m=24.18ns -> **min mclk freq. is 41.3MHz**

Case b.y will do it for an mclk frequency above 41.3MHz.

Note: If required, we could disable the h/w Icon and h/w Cursor when in full screen live video from Video Port.

Conclusion on Video Port Cheap System Solution

It is possible to achieve a minimum cost mother-board solution for full screen 640x480 30fps live video by interconnecting directly a digital TV decoder to Nordic Video Port under the following conditions:

1. **Prefetch only half CRT-FIFO instead of the entire FIFO**
2. **Do one refresh per line (eventually use extended refresh DRAMs)**
3. **To display 4:2:2 YUV use 32 CRT-FIFO stages and 16 TV-FIFO stages (between CRT-FIFO and MVA-Buffer we have enough storage now, but it has to be reconfigured). Minimum memory clock frequency is 48.9MHz**
4. **To display 4:1:1 YUV use 24 CRT-FIFO stages and 12 TV-FIFO stages. Minimum memory clock frequency is 41.3MHz.**

Under the same conditions, it is possible to display live video in a window with a 640x480 TFT panel, but the image has to be scaled-down and filtered with PX0070 or PX0072 - a more expensive solution. Also, during the actual live video display time, CPU and BLT will get very few cycles. This is not a factor with full screen live video as the CPU does not need to access the Video Memory at that time.

The question is now: how much cheaper is this solution relative to a full screen TV with overlay ? Sasha is working on a cheap solution with overlay.

Minutes of the Meeting on MM in Nordic in Nov. '93

With D.Keene, Keith, Del and Nordic Team on

- Video Port with 32bit memory is bwth. limitted. It is not a good solution for 32bit bus - better use Overlay or Sashapak.

4.6 On WavePort Architecture and Audio Driver

Starting from Dave Keene's WavePort Specification. Rakesh, Sasha and Vlad came-up with some ideas which could greatly simplify the design work for WavePort, reduction the complexity and the number of gates of the WavePort.
We'll start with the small issues and move towards more important issues.

4.6.1 AUXS1 and AUXS2 Pin

A. Dave proposed to share this pin with PDN (Power Down) WavePort Pin and configure it as AUXS2 when using 4231S.
We noticed that 4231S does have a PDN pin which we'd like to use to save power.
We'll put AUXS2 on another pin.

B. Dave proposed to have a fully programmable location for AUXS1 and AUXS2 CPU I/O map with a default at 530H and 388H. This is costly (4 registers and fast comparators) and very difficult to design at required speed: will slow down address range decode for I/O a lot affecting command to command parameters.
We suggest to have a fix address for these ports: 530H and 388H.
Two register bits (one per address) can disable the chip from answering to these I/O addresses.
If we are not sure where to place these ports, we can leave it as a metal option for future modification or have one more alternative for each, but the address should be fix for fast decode.
GR58:GR5B are no longer needed if this option is accepted. We'll need two bits to disable 530H and 388H address decode and eventually two other bits to select other two alternate I/O addresses. We should always decode 16 addresses: 530:53F and 388:397 (not 4 or 16).

4.6.2 WavePort R/W Buffer Location and Size

To simplify the h/w and reduce verification work-load , WavePort R/W Buffer Size should be fix: 32KB for each, enough for at least 180msec of operation -> 90msec for half buffer. Every 90msec the CPU will have to fill and/or empty half Audio Buffer (write or read respectively).
It is also possible to place the WavePort memory buffer in a fix memory location, common for both desktop and portable architectures. We could place it in Memory in the upper part
(in high address area),.just before the h/w cursor. If the h/w cursor takes the upper 16KB of the memory, we could use the next 64KB for the WavePort buffer.
Going down in memory, in Nordic we'd place the h/w icon in the next 16KB (only 4KB needed now... but leave some room for expansion) and the Half Frame Buffer for monochrome and color panels in the next 32KB and 96KB respectively.

Total:

96KB for CRT/TFT/SS STN panels,
128KB for monochrome panels and
192KB for dual scan color STN panels.
out of 1MB or 2MB memory available.

Cirrus Confidential Business Information

CL 152716

When the memory is expanded, all these fix components should move automatically in the upper side (towards the end of memory).

This places the WavePort buffer at:
- for **1MB** of memory by 32 (256Kx32 -> 00000:3FFFF) **3B000:3EFFF**
- for **2MB** of memory by 32 (512Kx32 -> 00000:7FFFF) **7B000:7EFFF**

If we adopt this solution, GR43 is not needed.

4.6.3 On Host Access to WavePort Buffer

In order to ensure independence between the Graphics s/w Driver and the Audio Driver, taking into account AVGA limitation of no CPU accesses during BLT operation, Dave proposed a separate path for WavePort Host Accesses. This involves a separate 4 stage by 32bit write and read buffer and separate arbitration for the Audio cycles. The buffer is shared by WavePort read and write operations and an I/O to an extension register bit is needed in order to switch from a read operation to a write operation or vice-versa.
Dave also proposed that the host address be dependent on GR6 CPU address map bits such that the video and audio address maps don't overlap.
Please note that this approach prevents debug with a Hercules card when in graphics.

In the following an alternate scheme is proposed:
- There is one register bit that enables WavePort memory write accesses and one register bit that enables WavePort memory read accesses. When one of these bits is asserted ,the chip expects only host accesses for the WavePort: the host address will be automatically translated to a memory address inside the WavePort buffer. The access will be fully linear, 32bit only, **graphics mode independent** (like in extended 8bit/pel). The Graphics block will be forced to this mode and all planes will be automatically enabled. On the Host side, the WavePort buffer will be placed always at A000:0000 - A000:FFFF -> 64KB with 32KB for read buffer (the upper side) and 32KB for write buffer (the lower side of this 64KB segment).
- The CPU Data Path is the same as for any other CPU cycle: no special FIFO, no special arbitration are required. The Audio Driver and The Graphics Driver communicate through interrupts: There is an interrupt for BLT completed (Dave's idea when presented with this proposal) and an interrupt for Audio Buffer Full (or Audio Transfer Completed). When the BLT is completed, if Audio is enabled on the chip, the BLT Completed interrupt allows the Audio Driver to check if there is any need to service the Audio and vice-versa. If possible, the Graphics driver should restrict the extent of the BLT operations to less than 160msec or so (leaving at least 20msec to start filling the WavePort buffer).

- There is no on chip CPU WavePort address generation. The CPU address in the range A0000:AFFFF (incremented by 4 every cycle for DW accesses) is translated to physical memory address 3B000:3EFFF. If host side WavePort pointers are used, the host address is latched and translated for comparison with the Codec pointers generated on chip.

Adopting this scheme would greatly simplify the extent of the design and verification work for WavePort: 8 weeks at least will be saved.

4.6.4 On the Audio Driver

Dave proposed that the Graphics Controller has h/w to compare the WavePort pointers and generate interrupts. The Codec pointers are I/O readable by the host as well as several buffer status bits (Audio-IN buffer Full, Audio-In buffer half-full, Audio-Out buffer half-empty, Audio-Out buffer empty,...).

Nordic-1M will support Dave's proposed interrupt driven interface.
In this case the only important difference for the driver will be that it will have to generate the buffers address and not rely on the h/w to do this (Dave suggests that CPU will write and read always the same address A000:0 or B000:0 and the actual address on the host side is generated on the chip - we sugest to change this and let CPU generate the address).

Because CPU has to wait untill the a BLT operation is complete and the BLT has to wait for Audio buffer to be full or the transfer completed, two more interrupts are needed: BLT operation completed and Audio transfer completed. This way the Audio Driver and the Video driver can communicate with each other.

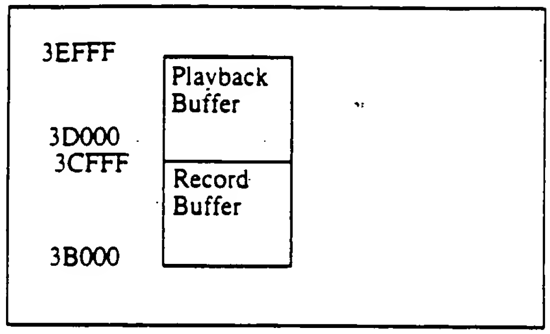
The Read and Write CPU side pointers will be generated by latching the last read and write WavePort CPU address (lower 16bits only) after translation to physical memory address. Comparators on the host and codec pointers will generate Buffer Full and Buffer Empty signals.

Jeff Ort drew our attention to the fact that the driver transfers a certain size block and needs an interrupt anytime half of what was placed in the Audio buffer was read by the Codec (the play-back buffer is half empty). To make things simple for the h/w, we propose to give Jeff an interrupt when the CODEC side playback buffer read pointer reached a preprogrammed value, value the driver programs everytime it transfers a block of a certain size in the playback buffer. For instance, if the driver put 2KB of data in the playback buffer starting from phisical address 3B000 (host address A000:0) and wants on interrupt when the first 1KB was read by the CODEC, it should program 3B0FF in the playback interrupt register, as 1KB = 256 32bit words. The register will have 16bits and the driver will program B0FF to get an interrupt when the CODEC reads 3B0FF or 7B0FF playback buffer address.

Similarly, on the record side, there will be a programable interrupt on the CODEC side to be used to tell the driver when the Record buffer is half full.
So we'll need two 16bit indexed registers for these programable interrupts. For each WavePort interrupt we also need a status bit to say that the interrupt happened, an enable bit to enable the particular interrupt. As Dave proposed, GR4D is the Interrupt Status register but bits [0] and [2] will be the programable interrupts status bits. The interrupts and the status bits will be cleared after this register is read (automatic clear in h/w).

We could use registers GR58:5B as high and low byte for the programable interrupt

on Record and Play-back buffers respectively.



The write-buffer (or Audio-Out =the one in which the CPU writes) is empty one Wave-Port write cycle to memory after

WRBUF-Write-Pointer = WRBUF-Read-Pointer+1

(this condition will activate set an empty flag).
The write-buffer is full one WavePort write cycle to memory after

WRBUF-Read-Pointer = WRBUF-Write-Pointer + 1

The same equations apply to the WavePort read-buffer, only this time the read-pointer is on the host side and the write-pointer on the Codec side.
Shouldn't be a Audio-Out buffer full interrupt and a status bit for it?

Threshold detect mechanism will require to increment host pointers, but the CPU is supposed to read this pointer and generate the proper address when going above the threshold. The idea of the threshold detector is that by incrementing the Audio-In read pointer every-time the write pinter is incremented if the Audio data is under the threshold value, the delta between the pointer remains the same and the buffer does not fill in with sub-threshold value. This was true if the half-full detection had been generated based on the delta between pointers. After talking to Dave, there are some clarifications to be made on the threshold detect mechanism for record:
For one record sesion, this is a one time event mechanism: if the threshold was passed once after being enabled, its effect is lost - even if the sound level goes below the threshold the record buffer read pointer works normally. The threshold will help only in the initial stage before the first actual voice pattern is detected, but it does not have any effect after the intitial triggering event - once the sound leved passed once the threshold.

With this driver approach, GR43 and GR58:5B will not be needed.
Five registers are saved. Twenty One register are still needed (plus two reserved = 23).

Another possible approach for the driver uses extensively the Vertical Interrupt:
In this case every Vertical Interrupt, the Audio Driver reads the Codec side pointers, compares them with the Host side pointers, the driver generates based on the CPU address it last wrote to or read from and makes its own decisions. The driver will also check the BLT completed status bit.

In other words, every 18msec (@60Hz), on the Vertical Interrupt the Audio driver can process data from the Codec pointers. calculate based on the last address if the WavePort buffer is full or empty and decide how many more WavePort accesses to execute. As at least 180msec worth of data is in a full Waveport Buffer. the driver has 6 opportunities to find out that the buffer is empty and start filling it before half WavePort buffer is empty.
On chip h/w is still needed to stop sending data to the Codec after sending all data and to read all data from the WavePort read memory buffer on the host side. Because the CPU decides when the buffers are full. half full or empty there is no need for registers for the pointers on the CPU side. but the CPU address will still be latched to generate the host side pointers.
The "BLT operation completed" and the "Audio transfer completed" interrupts are needed anyway bacuse we share the data-path for BLT and CPU cycles.

By adopting this mechanism in the driver, no special h/w is needed to preserve pointers on the Host side. to compare the Host side pointers (read and write) with the Codec pointers and to generate interrupts and status bits. The only interrupts needed are Standard VGA Vertical Interrupt as a time base. BLT completed Interrupt and Audio Completed interrupt.

We can still implement the threshold detect.

GR44,GR45, GR48 and GR49 won't be needed.
Nine registers are saved. Seventeen register are still needed (19 with two reserved registers).

Conclusion

The above proposal. if OK for the WavePort driver, would reduce significantly the amount of design and verification work on the WavePort. This document was written in order to get feedback and reach an agreement ASAP. Our goal is to have a unique s/w model for the WavePort. but we'd like to cut the amount of work if possible.

Rakesh's notes on the discussion with Jeff Ort and Dave Keene, 11/03/93

1. Since in Nordic, we do only 1R + 1P always, what happens if the bit is disabled for the host read at a time when ODD number of FIFO stages are filled? At Present we assume that while recording some SILENCE is inserted at the end of recording session and thus we can disable the bit as soon as driver disable it.
Jeff and Dave Kenne both agree that even though no SILENCE is inserted at the end, it is still valid to disable the memory transfer immediately after the bit is negated.

2. Threshold Issue.
If threshold mechanism is enabled, we keep putting the samples into the AUDIO BUFFER.
If the data in buffer is more than the programmed value (GR74 and GR75) and input data crosses the threshold value, the interrupt is generated immediately. But if data amount is less than the

value in the registers GR74-GR75 and input data has crossed the threshold value then we wait till it crooses this value(GR74 and GR75) and then interrupt is generated.

Dave's specs talk about two pointers on the RECORD buffer. one for the CPU read and another for CODEC write. Both of these pointers are accessable by the CPU in his specs. So he suggests that before the threshold is met we should fill up the buffer till half full condition. Once the buffer is half full, any more writes to RECORD buffer by the CODEC should increment both of these pointers. In this case whenever the threshold is met we are guaranteed to have some information about the history of audio samples just before the threshold is met. At the time when threshold is met the generated interrupt should tell CPU that CPU can read the data from the location pointed to by CPU read pointer.

But in NORDIC, the CPU read pointer is not accessible by CPU. Thus we keep putting the samples in the RECORD buffer and not generate the interrupt till the threshold is met even though we might pass the programmed value for the interrupt. At the time when the threshold is met, the interrupt is generated if CODEC write pointer is past programmed number. If threshold is met before the CODEC write pointer passes the programmed number, we wait till it does cross the programmed number. So the driver for Nordic should store the CPU address at the time when the threshold mechanism is enabled. So at the time of interrupt CPU can calculate the amount of samples in the BUFFER.

3. Independent control for IN / OUT mono / stereo .

Codec read and write are independently controllable for mono or stereo input / output.

4. Jeff agrees with the usage of OFFSET register GR9 to generate the CPU side adress for audio, instead of NORDIC generating the adress automatically.

Minutes of 11/20/93 Meeting with Keith, Del and Dave Keene

1. We need to check if WavePort accesses work with linear addressing (SR7 - aperture register).
2. Make sure that OverRun Status is there even if OverRun Interupt is disabled.
3. Enable looking for threshold only after you reach the given address.
4. Bit GR6E[6] should move to GR6E[3].
5. Talk about programable address for AUX1 and 2.
6. Check out arbitration for audio throughly.
7. Check overrun interrupt during threshold... always programmed interrupt after crossing threshold. ?? Rakesh may understand this!!

4.6.5 Registers specs for the AUDIO of NORDIC1-M.

GR60 Wave Port Control 1

7. Reserved
6. Reserved
5. Enable audio for host accesses.
4. Select 4215
3. Stereo for reciever (i:e STEREO CODEC)
2. Stereo for transmitter (i:e STEREO NORDIC)

1:0 Data Format select

00	16 bit
01	8 bit
10	4 bit
11	reserved.

GR61 Waveport control 2

7. Reserved
6. powerdown control ---> 0 will force low on PDN.
5. Enable command mode. 4215 must have been placed in COMMAND mode by forcing low on D/C* pin.
4. input level in SDI in command mode.
3. output level on SDO in command mode.
2. output level on SCLK in command mode.
1. output level on FSYNC in command mode.
0. output state of D/C* pin.

GR62 Input Audio Threshold setting.

7:0 Available only for 16bit mode. Only +ve data is compared i:e if msb of left data is 1 no compare is performed.

GR65

7. Enable Threshold detect mode.

6:0 Reserved

GR66 7:0 Audio Input Codec write pointer low byte

GR67 Audio Input Codec write pointer high byte

7. Enable input from codec to audio buffer.

6:0 Audio Input Codec write pointer (low 7 bits of 15 bit pointer)

GR6A 7:0 Audio Output Codec Read pointer low byte

GR6B Audio Output Codec Read pointer high byte

7. Enable output to codec from audio buffer.

6:0 Audio Output Codec read pointer (low 7 bits of 15 bit pointer).

GR6C : Interrupt Control.

7:6 Reserved.

5. Standard VGA Interrupt disable.
   0 --> Allow VGA interrupt
   1 --> Disable VGA interrupt.

4:2 Reserved.

1. Audio Input interrupt control -----> DURING RECORD SESSION.
0 --> disable this interrupt source.
1 --> Enable input buffer full OR programmed full interrupts.
0. Audio Output interrupt control -----> DURING PLAY SESSION.
0 --> disable this interrupt source.
1 --> Enable Output buffer empty OR programmed empty interrupts.

GR6D : Interrupt Status.
Indicates any active enabled interrupt. Reading this register will clear the interrupt state and reset the status. No effect on the standard VGA interrupt status or interrupt.
For All of the following 1 in the bit means PENDING INTERRUPT.
7:6 Reserved.
5. Standard VGA Vsync interrupt (state of 3C2(7)).
4. Reserved.
3. Record Buffer FULL (OVERRUN).
2. Record Buffer PROGRAMMED FULL.
1. Play Buffer EMPTY. (UNDERRUN)
0. Play Buffer PROGRAMMED EMPTY.

GR6E : AUX1 and AUX2 control
7. Enable Aux Sel 1 low-true output on adress match 530-53Fh.
6. Enable Aux Sel 2 low-true output on adress match 388-397h.
5:0 Reserved.

GR6F: Code status.
7:4 Reserved
3. Timer status (ADI) time slot 6 bit 7
2. overrange (OVR) time slot 7 bit5
1. PIO1 input status. time slot 7 bit 7
0. PIO0 input status. time slot 7 bit 6

GR70: Codec control DWORD from Nordic to codec. byte D7:0
GR71: Codec control DWORD from Nordic to codec. byte D15:8
GR72: Codec control DWORD from Nordic to codec. byte D23:16
GR73: Codec control DWORD from Nordic to codec. byte D31:24
GR74:75 Interrupt value for audio IN buffer.
When CODEC reaches the programmed adress -1, an interrupt is generated during the record session.
GR76:77 Interrupt value for audio OUT buffer.
When CODEC read pointer reaches the programmed adress -1, an interrupt is generated during the play session.

4.7 Four Bit per Pel Packed Format

To fully support Windows Chicago, Nodric-1M should support 4bit/pel packed format. CPU will be able to write only 8bits or more at a time. To modify only 4bits the CPU will do read modify write and mask four bits (bring them from CPU latches).
The following picture shows the actual format:

Memory Data

7 4	3 0	15 12	11 8	23 20	19 16	31 28	27 24
P0	P1	P2	P3	P4	P5	P6	P7
msb lsb	msb lsb	msb lsb	msb lsb	msb lsb	msb lsb	msb lsb	msb lsb
Pixel Left	Pixel Right						
1-st pel	2-nd pel	3-rd pel	4-th pel	5-th pel	6-th pel	7-th pel	8-th pel

In order to get Nordic-1M to support this mode, we need to place the VGA Video Data Path in 8bit per pixel mode with pixel doubling (mode 13) but disable the mechanism that concatenates adjaisant 4bits to make one 8bit pixel displayed twice (at a divided by two RAMDAC VCLK) and disable as well dividing by two the RAMDAC VCLK.
On the CPU side chain 4 address scrambling should be disabled and data should be shifted right like in extended 256 color modes.
In nordic-regs-rev9.lst we assigned SR26[0] to enable this graphics mode.

4.8 Robin's Ideas on High Resolution Panels for Nordic-1M

Nordic's objective is to be able to center and fill the screen on 800x600 panels and to center an 800x600 picture on 1024x768 panels. The main focus is 800x600 panel support TFT and dual scan STN.
In TEXT there will be three options:
- horizontal and vertical centering with 9x16 fonts (disable 640x480 force to 8dots font)
- auto-expand the font to fill the screen
- special font to fill the screen

In Graphics there will be tree options:
- horizontal and vertical centering
- use a driver to run at maximum resolution
- automatic expand to 800x600 (only if s/w verification shows it being OK).

In all cases horizontal and vertical options will be decoupled (different enable bits).

Here's a list of what needs to be done:
a. Expand horizontal and vertical centering capability from 640x480 to 1024x768.

b. Shadow all H/V CRTC registers except HDE and VDE.
The vertical registers are shadowed now, but the horizontal registers are not.
We need to shadow CR0, 2, 3, 4 and 5. These registers will be written by the BIOS at POST. The same mechanism to enable/disable vertical shadow is used to enable/disable horizontal shadow.

c. Support 10dot character-clock and CRT-FIFO-Read / Shift/Load (Video-Serializer Load). Now only 8dot and 9dot character-clocks are supported. For smooth-scrolling the pel-pan has to be expanded to 10 pisitions (0:9).

d. TEXT auto-expand will be as follows:
- horizontal: replicate the last pixel once for 9 dots wide fonts and twice for 8 dots wide fonts.
- vertical: replicate all odd scan-lines

TABLE 3. Font Expansion for Different Panel Sizes

Font-Size/Panel-Size	480	600	768
8x14	8x19	10x21	10x21 + center
8x8 with scan-ln. doubling	8x19	10x24	10x24 + center
9x16	8x19	10x24	10x24 + center

Please note that in text vertical expansion is done on the scan-line counter as well as the vertical display line counter, but it does not affect the vertical non-display line counter.

e. Graphics modes expansion (to be implemented only after s/w verification)
Graphics expansion is less important as it is assumed that important applications will run at maximum panel resolution with a driver.
We'll need to supply drivers for many applications with high res. panels !!

Vertical expansion:
- 350lines -> replicate all odd scan-lines
- 400lines -> replicate all odd scan-lines
- 480 lines -> replicate every 4-th line

Horizontal:
- center
- duplicate every 4-th pixel
- ~~duplicate every 4-th pixel with diagonal shift~~

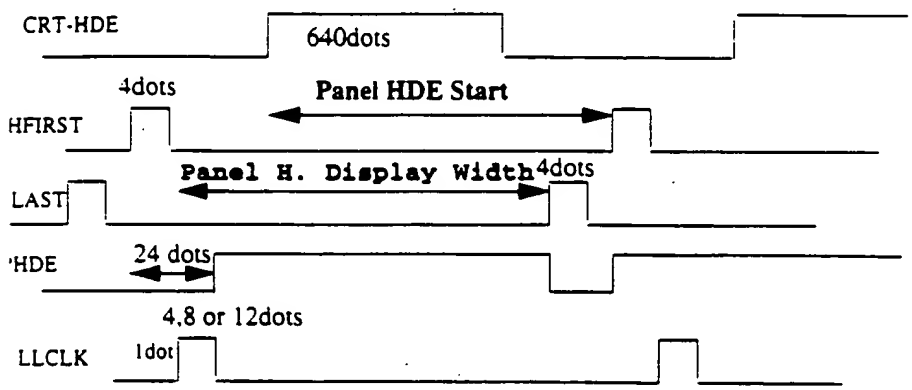
~~(scan-line 0 -> 1st pel, scan-line 1 -> 2nd pel, scan-line 2 -> 3rd pel, scan-line 3 -> 4th pel)~~

f. Please note that when character clock is expanded to 10 dots, h/w cursor also needs to be positioned at 8 or 9, so it needs one more bit of fine position.
SR2E[0] will be the msb of h/w cursor (default zero) to be used in this case.

4.8.1 Panel Horizontal Timing Control in Nordic

If in Terminator all panel timing is based on HSYNC and VSYNC programing, in Nordic it will be independent of HSYNC and VSYNC programming.

Horizontal Timing for 800x600 panel with 640x480 picture:



Registers for Panel Horizontal Timing Control

This set of four registers are used to generate horizontal panel timing with 640x480 or 800x600 panels and to center horizontally a 640dots or 720dots picture on an 800x600 panel.

CR40 No Center Panel HDE Start Register

[7:0] Panel Horizontal Display Enable Start relative to previous HDE start, in 4 dot-clkock units. The dot-clock used is never divided by two. This register is used to program Panel Line Clock Start and Panel Display Enable Start with both 640x400 and 800x600 panels any time no horizontal centering is needed.

CR41 Panel HDE Start Register to Center 720 dots

[7:0] Panel Horizontal Display Enable Start relative to previous HDE start, in 4 dot-clkock units. The dot-clock used is never divided by two. The value in this register

will be automatically used by Nordic to control Panal
Line Clock and Panel Horizontal Display Enable Start
anytime a 720 dot mode (text or graphics - RIX mode only)
is centered on an 800x600 panel. If horizontal expansion
is on CR40 is used.

CR42 Panel HDE Start Register to Center 640 dots

[7:0] Panel Horizontal Display Enable Start relative to previous HDE start, in 4 dot-clkock units. The dot-clock used is never divided by two. The value in this register will be automatically used by Nordic to control Panal Line Clock and Panel Horizontal Display Enable Start anytime a 640 dot mode (text or graphics) is centered on an 800x600 panel. If horizontal expansion is on CR40 is used.

CR43 Panel HDE Dot-Clock Skew Control Register

[7:6] Panel Line Clock Width:
- 0 = 4 dot-clocks
- 1 = 8 dot-clocks
- 2 = 12 dotclocks
- 3 = reserved

[5:4]CR42 value fine (dot-clock) skew:
- 0 -> no skew
- 1 -> delay Panel HDE start by one dot-clock
- 2 -> delay Panel HDE start by two dot-clocks
- 3 -> delay Panel HDE start by three dot-clocks

[3:2]CR41 value fine (dot-clock) skew:
- 0 -> no skew
- 1 -> delay Panel HDE start by one dot-clock
- 2 -> delay Panel HDE start by two dot-clocks
- 3 -> delay Panel HDE start by three dot-clocks

[1:0]CR40 value fine (dot-clock) skew:
- 0 -> no skew
- 1 -> delay Panel HDE start by one dot-clock
- 2 -> delay Panel HDE start by two dot-clocks
- 3 -> delay Panel HDE start by three dot-clocks

By programming this skew, it is possible to compensate internal delays on Panel HDE for diferent types of panels. If all delays are matched the skew fields should be zero to

align with character clock.

Automatic switching between these three registers will be done based on CRTC and Sequencer programming and the type of panel used
(640x480 or 800x600).

CR 44 Horizontal Panel Display Width Register

The value in this register is expressed in 4 dot-clocks (never divided by two).

[7:0] **Panel Display Width in hex expressed in 4 dot-clock chunks. As it is used to compensate for internal delays, this value is close but not exactly 640/4 or 800/4.**

4.8.2 Horizontal CRTC Registers Shadowing

In order to run VGA programs on 800x600 panels, Nordic needs to shadow the horizontal timing registers except the display enable end, which is application dependent.
(T1 has shadows only for horizontal total.)
This is because all horizontal timing registers but HDE have to be set-up for 800x600 in order to run an 800x600 panel. Symulscan with 800x600 panels runs even the CRT as an 800x600 display in all graphics modes.
In T1 as well as in Nordic, vertical panel registers are mapped at R2x, R3x,... accesible when CR2D[7] = 1. In nordic these registers remain there.
IN T1 there are two registers: **R0x and R1x** which represent HTot shadows, accessible with CR2D[7]. They are no longer used in Nordic, **but there is a replacement for them, independent of CR2D[7].**

In Nordic, H Total, H Retrace Start and End, H Blank Start and End are shadowed. Each entity has two values to chose from: a value for dclk/2 and a value for dclk not divided by two. The effect selection is done automatically based on SR1[3] value, but for access there is a selection bit: CR2C[4] to control which of the two sets of registers gets read or written.

Cirrus Confidential Business Information **CL 152728**

Here is a summary of the horizontal shadow registers and its controls:

CR2C[5] write protect for horizontal shadow:

1 ->. write both the shadow registers and the corresponding standard VGA registers fields (some registers are not accessed entierly but only in their timing part)

read back the shadow registers

0 -> write only the standard horiz. timing VGA registers (the shadow registers are write protected). Changing onlt the value of these registers does not have any effect over the CRTC timing.

read back the standard VGA horiz. timing registers

CR2C[4] Select the set of horizontal timing shadow registers to be used (low rez or normal). This bit is reutilsed from T1: it was low power RAMDAC mode - tie now the logic L) .

0 -> select Ry for access

1 -> select Rz for access

To diferentiate from the Rx registers which still exist as in T1, and are vertical panel registers, we'll call the two sets of horizontal shadows mapped in the same address space Ry and Rz (Ry = normal. Rz = dclk/2 case)

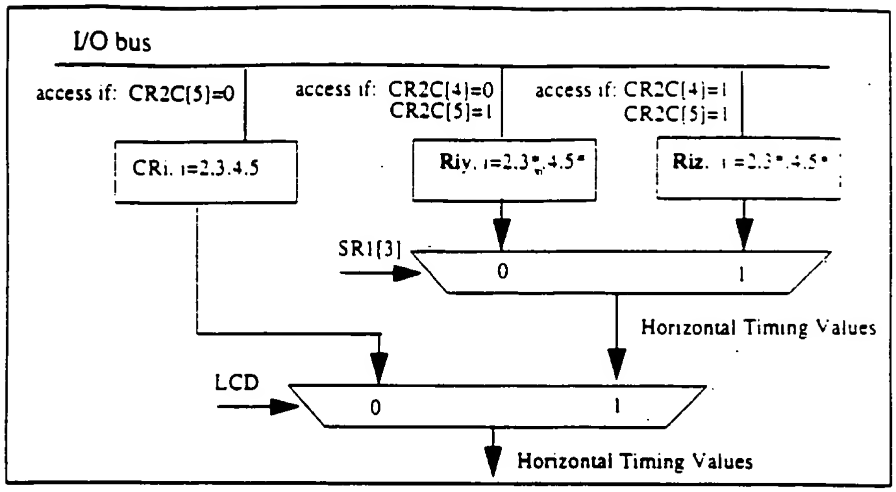
The effect of Ty and Rz is controlled by SR1[3] (dclk/2 bit).

The Horizontal Timing shadow registers are:
R0y,z R02y,z R03y,z (only lower bits) R04y,z R05y,z (only lower bits)

When CR2D[7] =1 we can access only the panel registers: R2x.... RBx. In this case we cannot access either the VGA standard registers or the horizontal timing shadow registers. **Please note that R0x and R1x are reserved in Nordic. Their function is replaced by R0y,z.**

Standard VGA CR0,2,3,4,5 horizontal timing registers (only the timing fields) do not have any effect on the CRTC: only the shadow registers have effect.

Note: Vertical Blank registers CR15 and CR16 are only write protected in T1 - not shadowed. This seems to be an error... so they should be shadowed in Nordic.
When using blank as overlay, DEn becomes blank and the shadow registers create the overlay both horizontally and vertically, but they are no longer write protected by CR2C[3] or [5]... As in 5428, they will not be write protected at all as the overlay is used only under MS Windows which is a controlled environment.



4.9 TV-Out Support in Nordic-1M

Nordic's objective is to be able output data to an analog TV Encoder chip thus supporting a cheap way of displaying on a TV. The main application is graphics, probably under windows, TV display and tape-recordig are of interest.
Nordic will run in a locked interlaced mode at 13.5MHz for NTSC and it will use the LCD panel supprt shadowing mechanism (the shadow Horizontal and Vertical CRTC registers) to prevent the application from changing the CRTC set-up. Even the dotclock frequency will be locked. Horizontal and vertical display enable will be open to the application.
Nordic will provide analog RGB, Composite SYNC, NTSC/PAL and TV-ON to the TV encoder. AD720 and MC1377 analog TV encoders that accept RGB input will be supported directly (with no glue logic).
Nordic will have edge skew h/w on the appropriate signals in order to meet the TV Standards timing requirements .
In this mode Nordic will not be able to run a panel or a CRT monitor... no simulscan supported with TV-OUT. But the same system will display on a panel, a CRT monitor or a TV with minimum glue logic.

Digital TV Encoders are not supported.

All standard VGA modes plus 640x480 8bpp can be supported, though special set-modes are required, as they all run interlaced. I suspect that we'll end up supporting only a few important modes. Should we compress text to 8dot wide fonts?

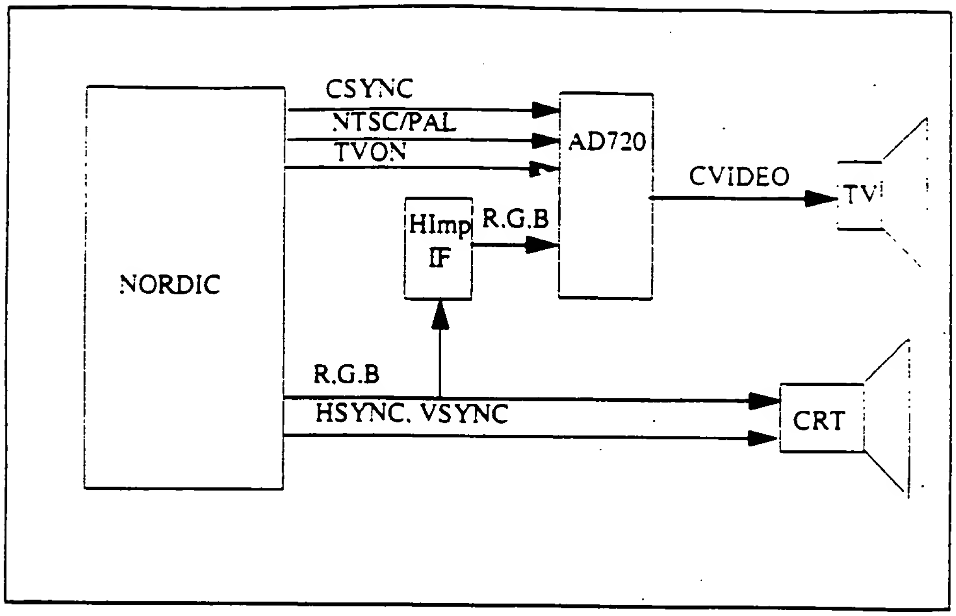


TABLE 4. Data on TV Standards

PIXCLK	Multiples of Line Frequency	Active Pixels per Line	Field Rate
12.2727MHz	780xFH	640	60Hz
14.75MHz	944xFH	768	50Hz
13.5MHz	858xFH	720	60Hz
13.5MHz	864xFH	720	50Hz

Signals to be supplied by Nordic:
- CSYNC = Composite SYNC signal
- NTSC/PAL = a programable output selecting the TV Standard NTSC (1) or PAL (0) The value of CR30[2] comes out on the pin.
- TVON = H for the TV Encoder to be ON, L for it OFF. The value of CR30[3] comes out on the pin. If CR30[3] =0, CSYNC and NTSC/PAL pins are L.

In Nordic-1M all these signals will come out to the panel pins when a certain configuration bit is enabled.

Vertical Timing

Vertical Blank and the appropriate CRTC registers are programmed to generate Blank which in TV case is the actual Display Enable.

For NTSC program 9 lines of VBlank.

For PAL program 7 lines of VBlank. In h/w Nordic will generate 7.5 lines of VBlank in this case by delaying the trailing edge by 0.5 line.

Vertical Sync Registers will be programmed for Field Sync generation.

For NTSC program 3 lines.

For PAL program 3 lines and get 2.5 lines by delaying the leading edge by half line.

Horizontal Timing

Two basic pulses are generated:

- HREF1 = Horizontal Reference 1 which is HSYNC start with 64 dotclocks width
CR30[1:0] controls the fine skew of this edge by increments of two dot-clocks.
- HREF2 = Horizontal Reference 2 which is a pulse marking the middle of the line between two HSYNC start edges. CR19[7:0] are used to program the start of this pulse.
CR30[1:0] controls the fine skew of this edge by increments of one dot-clock.

Note that HREF1 and HREF2 skews are in sync: HREF2 skew is half HREF1 skew.
Horizontal Total is 858 dot-clocks at 13.5MHz for NTSC and 864 dot-clocks at 13.5MHz for PAL.

Composite SYNC Logic Equation is:

CSYNC = [(HREF1 + HREF2) & VS] trigg368dclk-pls + -> Field Sync (serration)
[(HREF1 + HREF2) & VS*&VB] trigg32dclk-pls + -> Equalization
[HREF1 & VB*] trigg64dclk-pls -> standard sync during display time

Where: VS is Vertical Sync programmed time
VB is Vertical Blank programmed time
* is a logic negation
trig32dclk-pls means: on the rising edge of this signal generate a 32 dot-clocks wide pulse.

There are four ways to connect the video encoders and the CRT. We are now in the process of deciding which one is the best:

1. The simplest way to connect the TV decoder is to assume that the CRT Monitor should not be hooked to the notebook computer at the time the TV is. In this case we can put a high resistance voltage divider (2KOhm) to get the required 0.7V or 1.0V full swing to the decoder. The 2KOhm voltage divider will not practically affect the 50 Ohm resistance on the CRT, when the CRT is hooked on. The disadvantage of this method is that it requires a CRT to be disconnected from the note-book computer when connecting a TV. The TV encoders get RGB input through an AC connection (a capacitor) so the DC source may be a high resistance.

The advantage is that it is simple and low cost.

2. There are no buffers or other resistors than usual. We control the RSET resistor value on the LM334 current source depending on the presence of the CRT Monitor hook-up or not. If the CRT Monitor is not hooked-up, we adjust IRef so that DAC-s full swing output voltage is 0.7V (or 1.0V) with a 150 Ohm load. If the CRT Monitor is detected to be hooked up, RSet will be left as it is now. The adjustment is done automatically with a transistor. This approach requires a programmable output (from Nordic or Core Logic) to control RSet value. The detection of the CRT Monitor presence is done by s/w the standard VGA way.

3. Put 50 Ohm load on the DAC output and feed it to the TV encoder and to an Analog Voltage Repeater that drives the CRT Monitor RGB lines. This way DAC output is always 0.7V full swing independent if the CRT Monitor is hooked to the notebook or not.

This approach works for AD720 but not for MC1377 which requires 1V full swing. With this approach the presence of the CRT monitor cannot be detected VGA style (incompatibility issue ?).

4. Leave the CRT path unchanged, except for a low resistor shunt added in series.Convert the constant current delivered by the DAC into voltage that is amplified to NTSC/PAL level depending on the type of encoder used. Requires fast operational amplifiers on each gun, but allows operation with or without CRT connected and color compare.

This looks to be the best scheme but it is also the most expensive. The op-amp can be used also as a filter to limit the spectrum and avoid cross talk in color modulators.

Note: According to Fong-Jim Nordic DAC has some oprtational limitations: at 5V DACVDD, if the load is 150 Ohm and the DAC input is all ones (FFH) the output will be indeed 2.1V as expected.

If the load decreases (to more than 150 Ohm) , the output voltage will be increasing up to 2.7 V. At that point it will stop increasing: the DAC will be out of its operational range. This is OK for what we are trying to do with TV-OUT.

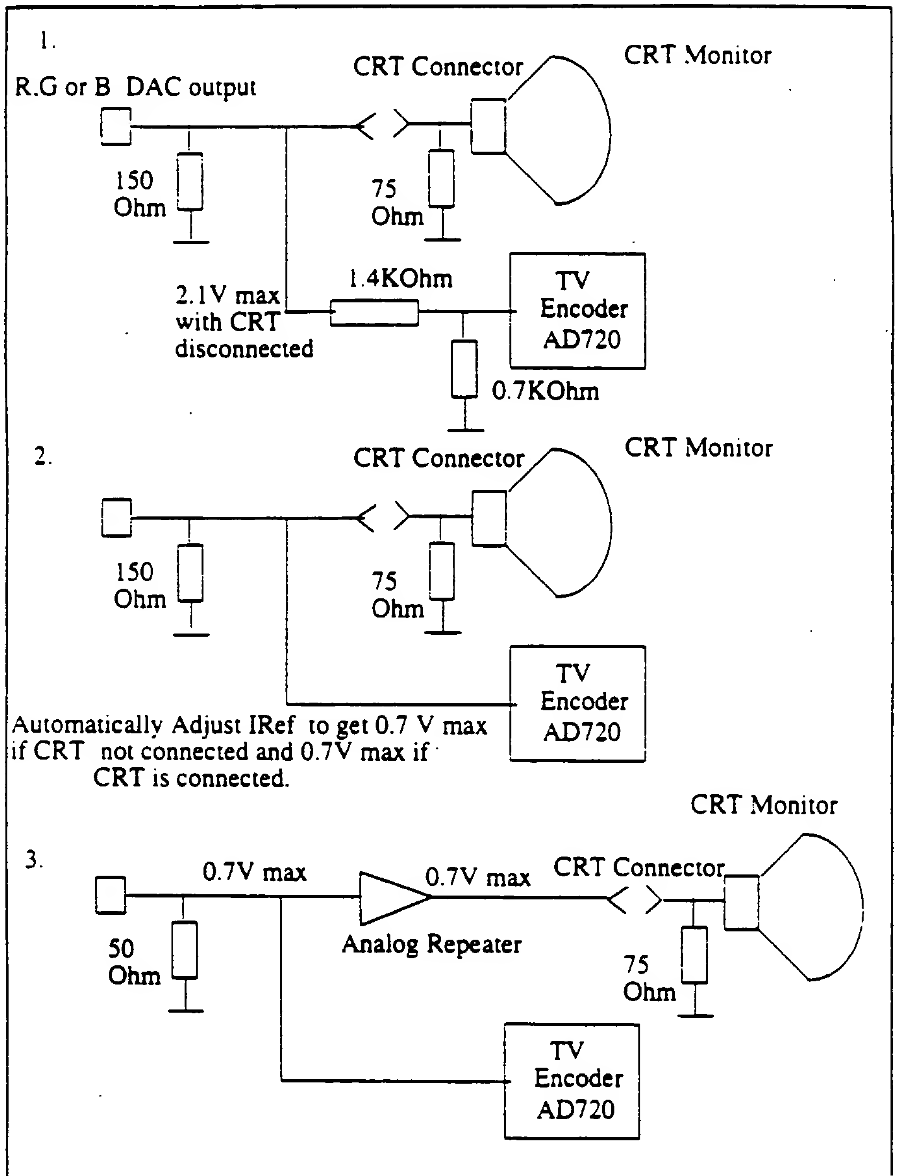
At 3.3V though, according to Fong-Jim the DAC output will follow the load only up to about 1.1V. This would require a 50 Ohm load at all times. With a 50 Ohm load the maximum DAC voltage at both 5 and 3.3V is 0.7V, within the operation range.

The question is: how to add a 75 Ohm load when there is no CRT in there?

I see two simple solutions: either use TV-OUT at 5V DACVDD (switch CVDD and DACVDD to 5V for TV-OUT) or make an external connector similar to a CRT connector that one plugs into the back of the notebook when the CRT is not in use. This connector should have 75 Ohm resistors to GND on RGB lines.

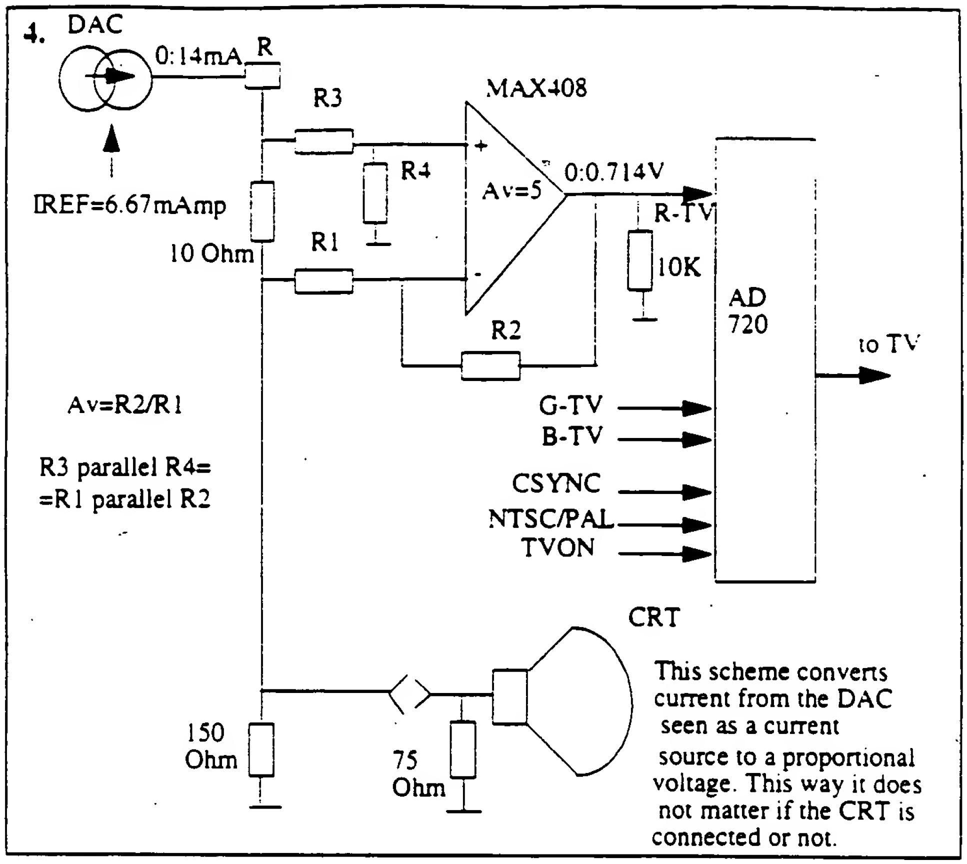
There may be other solutions too... everyone is invited think.

Three possible ways to connect an Analog PAL/NTSC TV Encoder to DAC output.



Cirrus Confidential Business Information

CL 152734


4.
DAC
0:14mA
R
IREF=6.67mAmp
10 Ohm
R3
R4
R1
MAX408
Av=5
0:0.714V
R-TV
10K
R2
AD
720
to TV
Av=R2/R1
R3 parallel R4=
=R1 parallel R2
G-TV
B-TV
CSYNC
NTSC/PAL
TVON
CRT
150
Ohm
75
Ohm
This scheme converts current from the DAC seen as a current source to a proportional voltage. This way it does not matter if the CRT is connected or not.

4.10 H/W ICON Support for Nordic-1M

Nordic-1M will support a h/w icon with the following features:
- Up to 4 64x64 Icons displayed at the same time as a vertical bar.
- All Icons move together and they are inedex by 64 scan lines.
  The Icon start reference point is top left corner.
- Horizontal Resolution is one pixel programed coarse as character clocks and fine as pixels exactly as a h/w cursor.
- Vertical Resolution is one scan-line.
- Each Icon has inependent display control for
    - icon number i enable for display (i=0:3)
    - display mode: 3colors + transpency or 4 colors
    - blink enable at text cursor rate (or half of it)
    - horizontal pixel doubling (extendes towards right)
    - vertical scal line doubling (extends down)
    - mamory map selection: two maps available per icon
- H/W Cursor goes on top of H/W Icon
- Icon memory model is and phisical address space are very similar to h/w cursor.
  Actually the icon resides in memory in h/w cursor area: Nordic supports half the number of h/w cursor maps 5428 does and uses the remaining space for icon maps. Eight 64x64 icon maps are supported: two for each icon we can display. Icon memory map assignment is fix.

<u>H/W Icon S/W Model</u>

Icon position will change at set mode if it is not compensated. BIOS should service the icon so that it keeps the position stable in all graphics modes: read current position and translate in the same position in the new graphics mode. The BIOS should have a call to set-up the h/w icon position and another call to control which icon is displayed and how.

What we need:

a. Horizontal Start Position in character clocks and pixels
  Will be programmed exactly as h/w cursor at SR10,30,50,70,90,B0,D0,F0.
  (or SR "odd" "zero"). The upper 3 bits of index define the fine (vclk) position.
  SR2A[6] is the fine postion msb to be used with 9 and 10dots fonts.See 542X Reference Manual page 9-13, SR10 description.

Note: 1. If character clock or vclk change (8/9/10 dot cclk or vclk/2) the horizontal position should be adjusted.
  2. As with h/w cursor the horizontal and vertical position of the icon are synchronized: s/w writes the horizontal position first, but the new position effect takes place only after vertical position was updated.

b. Vertical Start Position in scan lines.
  Will be programmed exactly as h/w cursor at SR11,31,51,71,91,B1,D1,F1.
  (or SR "odd" "one"). The upper three bits of index define the low order three bits

of position in scan lines. See 542X Reference Manual page 9-14. SR10 description.
How will Nordic know to program icon or cursor?
One bit will tell if icon is programmed instead of cursor. As the icon position is programmed seldom, this mechanism does not affect performance while saving address decode h/w.

c. Enabling icon position modification by CPU is done in the same register h/w cursor is enabled: SR12.

SR12[3] Enable CPU access for Icon Position Modification
(default:0 = disable = use the same I/O address for cursor position modification).
This bit affects both horizontal and vertical position.

d. Icon and h/w cursor memory map selection:
5428 uses SR13[5:0] for 32x32 h/w cursor memory map selection (64 maps) and SR13[5:2] for 64x64 h/w cursor memory map selection (16 maps).

In Nordic, SR13[4:0] still select h/w cursor memory map, but the memory address bit which in 5428 is controlled by SR13[5] is no longer controlled by SR13[5]. It is controlled by an internal signal whose value is always 0 when a h/w cursor is addressed and 1 when icon is addressed. So in Nordic SR13[5] is no longer used to select a h/w cursor map. Only half of the 5428 h/w cursor memory is available for Nordic h/w cursor memory map. In Nordic there are 32 32x32 and 8 64x64 h/w cursor memory maps available.

The icon address is generated exactly as the h/w cursor address, but the source of the address map selection is no longer SR13[5:0] (it cannot be as SR13 is used for h/w cursor). SR13[5] is also noy used. To generate icon mamory map select, a two bit counter counts the first, second, third and fourth set of 64 scan lines on the Vertical Total Line Counter (the one not affected by vertical expansion). This two bit counter output, called the Icon Index Counter (ICIXCTR) will replace SR13[4:3] in Icon Memory Map Address generation.
A signal called ICON-MAP will be generated during Icon addressing. This signal is 1 if an icon data is to be fetched from memory and 0 otherwise (including when h/w cursor data is to be fetched from memory).
ICON-MAP signal will replace SR13[5] in both h/w cursor and icon memory map selection address.
One register bit in each icon control field will select the first or second memory map for each icon to be retrived from memory. This bit called Icon Memory Map (IMM) will

replace SR13[2] in icon memory map generation.

5428 64x64 h/w cursor address,SR13[5:2] part:

b5 ↓ b4 ↓ b3 ↓ b2 ↓

Nordic 64x64 h/w cursor address. R13[5:2] part:

ICON-MAP=0 ↓ b4 ↓ b3 ↓ b2 ↓

Nordic 4x64x64 icon address, replacement for R13[5:2] part:

ICON-MAP=1 ICIXCTR[1] ICIXCTR[0] IMMi

where ICIXCTR[1:0] is the Icon Index Counter
IMM is the Icon Memory Map bit in each icon control register
ICON-MAP is a signal that is high for icon address and low otherwise

e. **Icons Control Registers**

Each Icon has a field of bits attached to it

0.- icon i enable for display (i=0:3)
1.- display mode: 3colors + transpency or 4 colors
2.- blink enable at text cursor rate (or half of it)
3.- horizontal pixel doubling (extendes towards right)
4.- vertical scal line doubling (extends down)
5.- mamory map selection: two maps available per icon

Six bits per icon, all defaulting to zero. The icon enable bits select which icon is displayed
Start at SR2A register.
Use **SR2A,2B,2C,2D[5:0]** for Icon 0,1,2,3 control.
If an icon is not enabled, do not generate the memory cycles for its data... if it is not difficult to do. and do not display that icon.

f. SR2A[7] is Icon Test bit. Normally it is zero, only in test mode it will be 1.

g. The colors for the Icon are in the 16 location RAM extension at location 258.
See SR12 description in 5428 Technical Reference Manual.
Locations 256 and 257 are accessible with pixel address 0 and F for h/w cursor access.
Locations 259, 260, 261 and 262 are accessible with pixel address 3,4,5,6 as icon colors 0,1,2 and 3.
When three colors plus transparent attribute is used, the pixel addresses will be 4,5,6, for colors 1,2,3 (so pixel address 3 and color 0 of the 16 location special RAM are not used).

h. **SR2A[6]** is the most significant bit of fine horizontal position for the h/w icon:

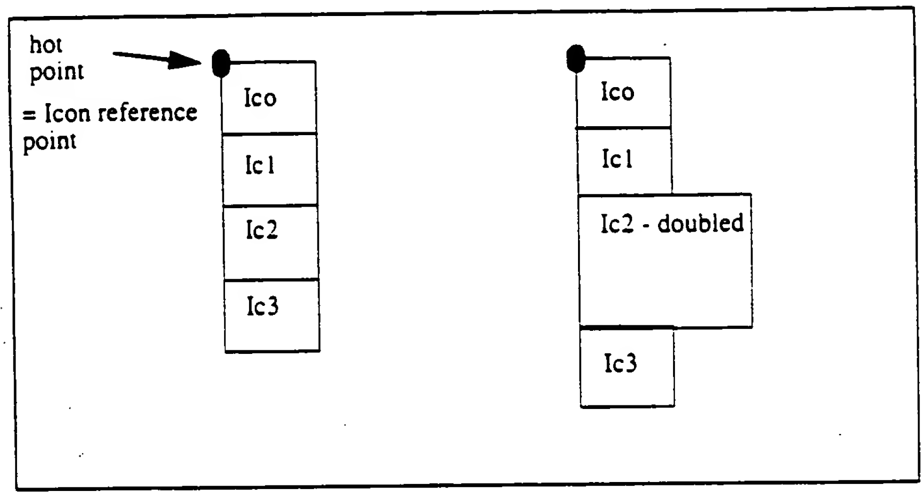
SR2A[6] SRX[7] SRX[6] SRX[5] --> bit3 bit2 bit1 bit0

Please note that SR2A[6] will be used only in 9 or 10 dots text when the fine position is 8 or 9.

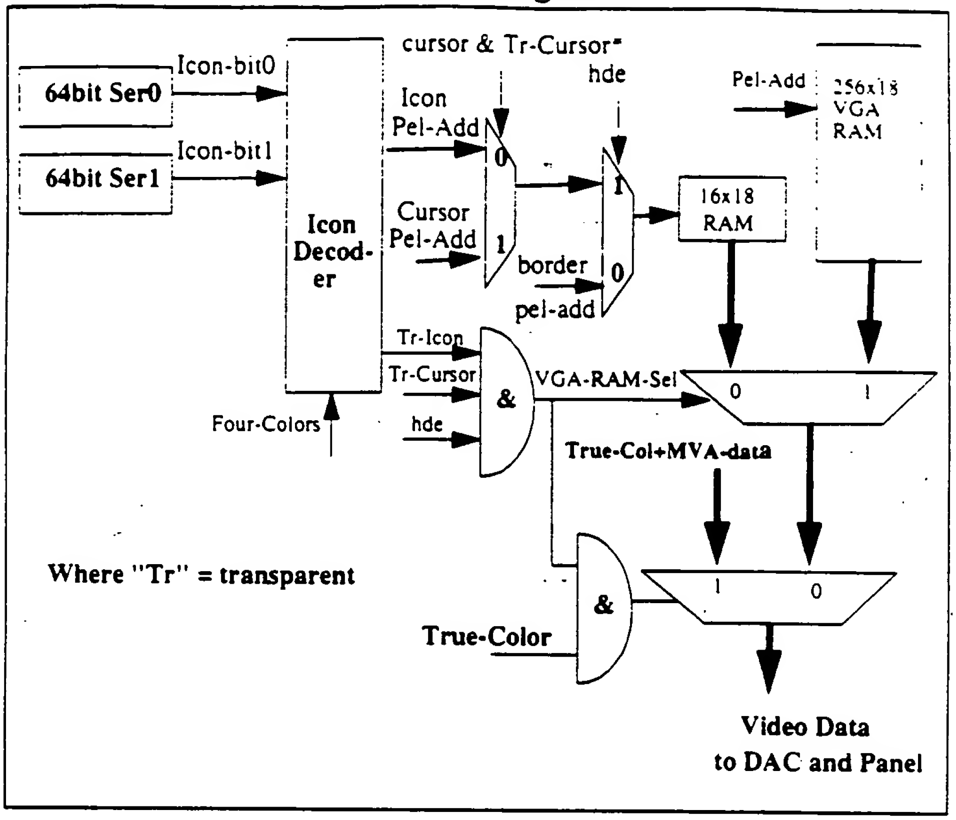
i. Please note that when character clock is expanded to 10 dots, h/w cursor also needs to be positioned at 8 or 9, so it needs one more bit of fine position.
SR2E[0] will be the msb of h/w cursor (default zero) to be used in this case.

H/W Activity for Icon

1. Horizontal and Vertical Window
2. Address Generation including the independent scan-line counter for Icon
3. Memory cycle generation (at the end of line, after h/w cursor if icon and cursor are on the same line)
4. Icon Latch, Serializer and Data Path including icon blink
5. Horizontal and Vertical Doubling



Icon Data Path Block Diagram



4.11 Dithering in Nordic

Due to new pixel depth modes and for increased flexibility the dither block in Nordic is diferemt than T1.
In Nordic the dithering is subtractive, (not additive like in T1). This moves the dither error (= double shade) to black where it is less obvious than it was before (when it was white).

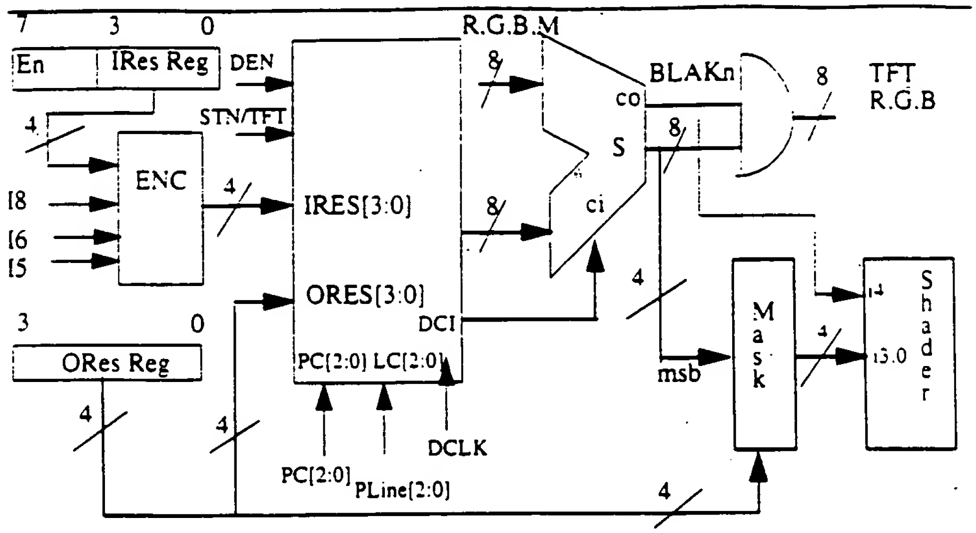
CR4C - Input Resolution Override Register for Dithering

Bit	Definition
[7]	Enable Input Resolution Override (default disable) Normally the graphics mode resolution is decoded and fed to the dither block. For all graphics modes that go through the RAMDAC RAM - 6bits/gun is automatically the input resolution. For 5:5:5 RGB or 5:6:5 RGB or 8:8:8 RGB we use

	a diferent input resolution: 5 for 5:5:5 and 5:6:5 and 8 for 8:8:8 RGB In all cases the proper input resolution is automatically selected. If there is a need to overide this value, than we wiill use this bit to enable the override and write the new value in bits 3:0. For 8 bit per gun we write 8 (1000), for 6 bits per gun -> 6 (0110) and for 5 bits per gun -> 5 (0101).
[6:4]	Reserved
[3:0]	The binary value of the override input resolution. 1000 -> 8bit/gun 0110 -> 6bit/gun 0101 -> 5bit/gun

CR4D - Output Resolution Register for Dithering

Bit	Definition
[7:4]	Reserved
[3:0]	The binary value of the output resolution for dithering 1000 -> 8bit/gun (TFT) 0110 -> 6bit/gun (TFT) 0100 -> 4bit/gun (TFT or 16 shades STN) 0011 -> 3bit/gun (TFT or 8 shades STN) 0010 -> 2bit/gun (4 shades STN) 0001 -> 1bit/gun (2 shades STN)



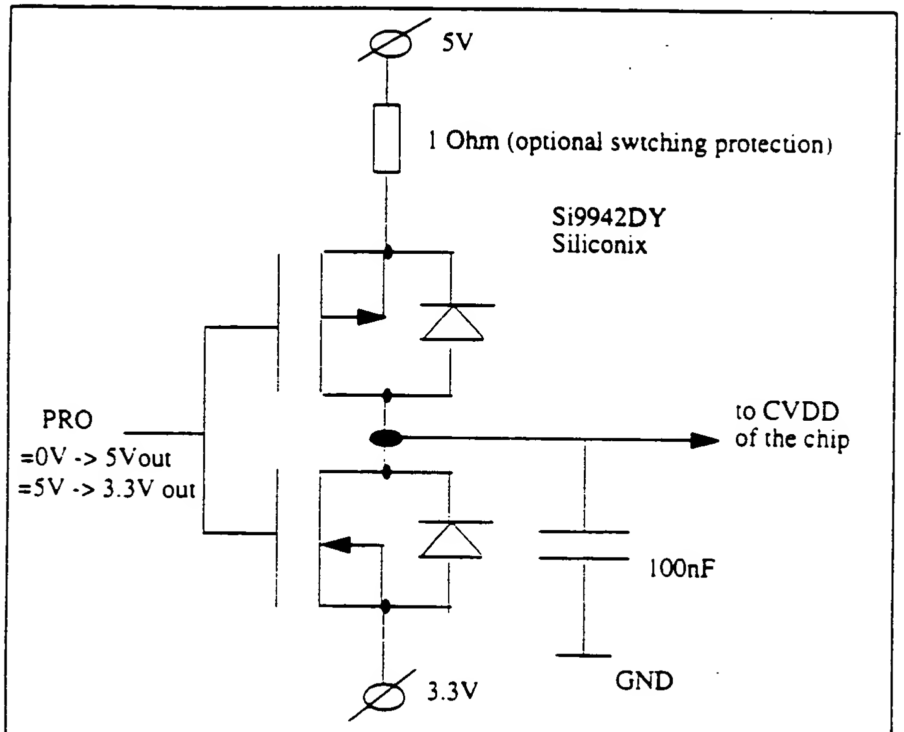
TFT: 8,6,4,3 bits/gun

STN: 4,3,2,1 bits/shade (16+1,8+1,4+1,2+1 shades, +1 = force black)

Nordic Dithering

4.12 Core-VDD Control

Because Nordic can run higher frequency memory-clock at higher Core VDD (5V) but it offers maximum power savings at low Core VDD (3.3V), we look for a mechanism to dinamically switch at set-mode or when going into suspend from 5V to 3.3V and vice-versa.
This would allow for instance to run 24bpp 640x480 mode at 5V CVDD while switching back to 3.3V when going into suspend mode or when switching to a 4 or 8bpp mode.
All Nordic needs to do in this case is supply a programable output to be used to switch an external circuit.



SEL3.3VI = Suspend + Reg-Bit
Use PAL/NTSC pin and CR30[2] as power control pin.when TV-OUT capability is not ever used in the system.
Use PRO pin and CR30[7] as power control pin, when TV-OUT is used.

The value written in CR30[2] always reflects on the NTSC/PAL pin which thus can be usd as a generic programable output.
In a system without TV-OUT capability, NTSC/PAL bit and its output pin can be used to

control when CVDD is 5V or 3.3V. This way, in most modes the chip may be optimesed for power running at a lower mclk frequency, while in certain modes it will be optimised for performance, running at 5V CVDD and a high frequency mclk (higher than what can be run at 3.3V). Because NTSC/PAL pin is tied to CRTVDD which is 5V in most systems, this bit is the first choice for the CVDD switching function.

5.0 Nordic-1M Pinout Specification 208 pin package:

System Interface (reference is VESA VL bus): **74**

(on BVDD)

DAT31:0
HIMEM0/ADR[24] (normally used as ADR[24], but it can be used as any upper address or any logic combination of upper addresses. The valid value of this pin for Nordic address space is programable in VESA VL and PCI bus.
HIMEM1/ADR[25] (normally used as ADR[25], but it can be used as ADR31 and leave 2GB of upper address space reserved for MPEG/JPEG boards). The valid value of this pin for Nordic address space is programable in VESA VL and PCI bus.
With these two pins, Nordic can directly decode up to 64MB of CPU address space in VESA VL bus.
ADR[23:2]
BE3n,BE2n,BE1n,BE0n
LADSn,LDEVn,RDYn
M/IOn, W/Rn
RESETn, LCLK
RDYRTNn,
EBROMn/SLEEPIn (Enable BIOS ROM Buffers on all busses or when not in ISA or SLEEP Input active L. If this pin is L, the chip goes in sleep mode as if 3c3[0] is 0 or 46E8[3] is 0. The ISAPU read-back register bit configures the pin for EBROMn. If not in ISA bus, this pin is configured as SLEEPIn and it should be tied H if not used.
INTR, CLK32K, OSC/XVCLK (14.318MHz or External VCLK)
SUSPI/BLI, SBYI/ACTI/FCEVIDEOn
(32KHz clock input,
Supend Input/ Back-Light Input
Stand-by Input/Activity Input/VAFC External Video Control Input - controlls the direction of FCP[7:0]. Pin configuration controlled by VAFCPU. Most systems will use only one direction for VIDEO, probably video in, so the fact the FCEVIDEO is on BVDD should not require voltage translation).

Configurations: - VL/PCI/ISA bus. support/not BIOS. 8/16bit BIOS.
- Need a register bit to select SUSPEND Input or Back-Light Input. Default is SUSPEND Input.
- Need a register bit to select Stand-By Input or Activity Input. Default is Stand-By Input.
- SUSPIn. BLIn. SBYIn. ACTIn are level triggered. active L inputs. In Nordic there is no Suspend pin polarity bit as in T1 (SR8[4]).

Note: a. In PCI bus. Alpine used ADR17:2 as BIOSA15:0. We assume that all portable systems will use a motherboard video BIOS and do not support an adapter BIOS in PCI. We do support an adapter BIOS in VESA VL bus and ISA to facilitate demo board design.

b. EBROMn is active on all bus types: in ISA is generated on Address Decode and MEMRn. while in VL and PCI is unlatched address decode. In PCI bus the BIOS should be relocatable. an option we do not support. That is why we say that we do not support BIOS in PCI. We do support IOCHRDYn and RDYn/TRDYn for BIOS accesses too. All this logic should already be in 5428 (ISA and VL) and Alpine (PCI).

c. The 5428 Feature Connector pins are available in all buses. A register bit enables them early in the logic to save power when not in use.

d. In ISA RESETn has to be inverted outside the chip.

System/Bus Interface Signal Mapping

(See 54xx Reference Manual for Feature Connector Pins Description: FCyyyy)

TABLE 5. VESA-VL <-> PCI bus <-> ISA bus Pin Mapping

VESA-VL	Intel-PCI	ISA
HIMEM	--	--
ADR23:16	--	LA23:16
ADR15:9	--	SA15:9
ADR8	PAR	SA8
ADR7	STOPn	SA7
ADR6:2	--	SA6:2
BE3n	C/BEn3	SA1
BE2n	C/BEn2	SA0
BE1n	C/BEn1	SBHEn
BE0n	C/BEn0	RFRSHn
DAT31:22	AD31:22	--
DAT21:19	AD21:19	--
DAT18	AD18	MWRn
DAT17	AD17	IOCS16n
DAT16	AD16	IRQ
DAT15:0	AD15:0	SD15:0

TABLE 5. VESA-VL <-> PCI bus <-> ISA bus Pin Mapping

VESA-VL	Intel-PCI	ISA
RESET	RESET	RESETDRV ??
LADSn	FRAMEn	BALE
LDEVn	DEVSELn	MCS16n
RDYRTNn	IRDYn	AEN
W/Rn	IDSELn	IORn
M/IOn	--	MRDn
LCLK	CLK	IOWn
RDYn	TRDYn	IOCHRDY
EBROMn/VADRn	VADRn	EROMn/VADRn
INTR	INTR	ZEROn

Notes: 1. SW2, SW0/MCLK/XMCLK and SW1 pins are normally used by the BIOS as panel type switches readable on SR24[2:0].

2. XMCLK and XVCLK (on OSC/XVCLK) are needed for manufacturing test of the chip. When they are used, SR24[1:0] will wiggle at clock rate and they should be masked on the tester.

3. MCLK and VCLK are needed for manufacturing test of the clock synthesizer and for chip debug.

DRAM Interface (for assymetric, dual WEn 256Kx16 DRAM-s): 50
(on DVDD)

MA9:0
MD31:0
RAS0n,RAS1n,CASn/WEn, OEn
WE0n/CAS0n,WE1n/CAS1n,WE2n/CAS2n,WE3n/CAS3n

Configurations: symetric/assymetric DRAM, dual WEn/CASn

Note: a) For dual CASn DRAMs, the WE0n:WE3n pins become CAS0n:CAS3n, while CASn becomes WEn.
b) Two RAS-es are used for two banks.
c) OEn is not actually needed as we use early write.

Clock Synthesizers: 6
(MAVDD, MAVSS feed mclk clock synthesizer, VAVDD, VAVSS feed dclk clock synthesizer)

MAVSS. VAVSS. MAVDD. VAVDD.
MFILTER. VFILTER
(Can we design a clock synthesizer without an external filter?)

CRT & TV-OUT Interface and RAMDAC related pins: 13
(DAVDD. DAVSS feed all RAMDAC custom layout)

R.G.B (analog)
IRef. DCVDD1. DCVDD2. DCVSS1. DCVSS2

On CRTVDD:
HSYNC , VSYNC TO (these two pins are on CRTVDD)
CSYNC O (Composit Sync to be used with any analog TV Encoders like AD720. CSYNC is activated by SR25[4]=1, otherwise is forced 0.)

NTSC/PAL O
(A programable output that can be used to control a TV Encoder input that selects the TV standard. This pin is a generic programable output. Its value is not affected by SR25[4].)

On FPVDD:
TVON/RES O (Generic Programable Output controlled by CR30[3] . It can be used to control AD720 TV Encoder Power Control Pin.) TVON is on FPVDD (not on CRTVDD like the other TV-OUT pins) This pin is also reserved for future use (DRAM-s).

Note: If TV-OUT feature is disabled CSYNC, NTSC/PAL and TVON are forced L (to avoid powering up the TV Encoder in case it is powered off when not in use).

FLAT PANEL and Power Management Interface (on FPVDD): 31

R7/SUD3, R6/SUD2, R5/SUD1, R4/SUD0, R3/SUD7, R2/SUD6, R1/FCP[3], R0/FCP[2],

G7/SLD7/sud3/UD3/TD7, G6/SLD6/sud2/UD2/TD6, G5/SLD5/sud1/UD1/TD5, G4/SLD4/sud0/UD0/TD4, G3/SUD5, G2/SUD4, G1/SUSPSTn/FCP[1], G0/SBYSTn/FCP[0]

B7/SLD3/sld3/LD3/TD3, B6/SLD2/sld2/LD2/TD2, B5/SLD1/sld1/LD1/TD1, B4/SLD0/sld0/LD0/TD0, B3/MOD, B2, B1/OVRW, B0/FCVLK

(use slow edge pads and stagger them modulo 4)

Cirrus Confidential Business Information

FPVDCLK. LLCLK. LFS. FPDE. (use slow edge output pads)
FPVCC. FPBL. FPVEE

Notes:

b. R1:0. G71:0. B1:0 (24 bit TFT panel extra video data out pins)
c. We need also bits for SUSPST and STANDBYST, PANEL-ON/OFF, PANEL-IN-POWER-UP-or-DOWN. Dwarka, do we have this now ?
d. We assume that only STN panels. need external MOD.
e. No NPD pin.
f. FCP[3:0] pins are I/O-s whose direction is controlled by FCEVIDEOn in VAFC configuration. FCVCLK is the clock that clocks them in and which should be used to clock them while they are displayed or at least to resynchronize them before being latched on the internal clock. FCVCLK is always an input when in VAFC configuration.

Nordic TFT Panel Data Pin Name versus the actual TFT Panel Data Signal for diferent types of TFT Panels

Nordic Pin Name-> /Panel Type	RGB7	RGB6	RGB5	RGB4	RGB3	RGB2	RGB1	RGB0
8bit/gun	RGB7	RGB6	RGB5	RGB3	RGB2	RGB1	RGB0	RGB0
6bit/gun	RGB5	RGB4	RGB3	RGB2	RGB1	RGB0	N.U.	N.U.
4bit/gun	RGB3	RGB2	RGB1	RGB0	N.U.	N.U.	N.U.	N.U.
3bit/gun	RGB2	RGB1	RGB0	N.U.	N.U.	N.U.	N.U.	N.U.

Where N.U. = Not Used for Panel Data with this type of panel.

~~Audio Port (on ADVDD): 6~~ **Removed on 12/24/93**

~~SDO (SErial Data Output to CODEC pin SDIN)~~
~~SDI (Serial Data input from Codec pin SDOUT)~~
~~FSYNC (Frame Sync Output)~~
~~SDCLK (Serial Data Clock I/O)~~
~~D/Cn/AUXS1 (Data/Control Select Output or CS4231S Chip Select for the ISA bus)~~
~~PDN (Power Down Output)~~
~~Note: These pins can be used as extensions for 24 bit TFT panels, in which case the chip does not support the audio port. **Note also that CS4215 has Resetn pin.**~~

Miscellaneous Pins: 12

Cirrus Confidential
Business Information

Following pins on MVDD:

CL 152748

SW2/RES4CS I (Switch #2 or reserved for SDRAM support in the future)
SW1/RES4CKE I (Switch #1 or reserved for SDRAM support in the future)

SW0/MCLK/XMCLK I/O with PD controlled by XReset
(Switch #1 or mclk driver buffered output for tesing and debug or external mclk for testing; a special register bit disables this output in normal operation for power savings.
Default: Input on which either SW2 or XMCLK comes in.
Only if the special register bit
is set, this input becomes output and outputs MCLK from the clock synthesizer for testing.
The special register bit defaults 0 = the pin is configured as Input. Please note that a h/w configuration PU/PD controlls if the actual internal mclk comes from an internal or external source, but this PU/PD does not control if SW2/MCLK/XMCLK is an input or an output.)

RES4PMS Free Pin with no buffer on it.

<u>Following Pins on PVDD:</u>

TRWn I (Test Latch Load Enable, active L - for pinscan & test mode)
VCLK/FCDCLK I/O (Switch #0 Input or vclk driver buffered output for testing and debug, or VAFC Dot-Clock Out ;
Default for this pin is Input on which SW#0 is applied.
VCLK and FCDCLK is the same signal, the name is diferent to ease reference to VAFC spec)
FCP[7:4] I/O Feature Connector Data Pins bits 7 to 4.
The other Feature Connector pins are shared with some panel or host bus pins.
FCESYNCn/RES4SDCLK I 5428 Feature Connector ESYNCn pin - controlls the direction of FCBLANKn and its effect and tri-states HSYNC and VSYNC.
FCBLANKn I/O Feature Connector active low bidirectional Blank signal. Its direction is controlled by FCESYNCn.

<u>Digital, Mixed Voltage Power Pins: 22</u>

BVDD1, BVDD2, FPVDD1, FPVDD2, MVDD1, MVDD2, CRTVDD,
CVDD1:4,
VSS1:11

Total Pins Used: 208 (with two Reserved pins and three other pins that could be freed.).

<u>Pinout Modifications in rev. 3.8 relative to rev. 3.7 09/11/93</u>

1. Took-out Whisper Support Pins: WWRn. WRdn. WAD[3:0]. FPCn.

2. Reduced the number of CPU Address Pins: ADR[24:27] went-away. Use HIMEM to place the chip outside 16MB of memory for linear addresing.

3. Killed CRTVDD and placed CRT Interface on Host Bus VDD.

4. Introduced VAFC pins in a unique pinout configuration in both VESA VL and PCI.
Added VAFCPU a pull-up to enable VAFC (VESA Advanced Feature Connector).
ADR[27:24] VAFC pins, and R[7:4], G[7:6] , EROMn/VADRn. SUSPI/BLI. SBYI/ACTI get a configuration for VAFC -> total 14 pins.
SW0/VCLK will be used as VAFC VCLK. losing SW0 pin as panel type when VAFCPU is used.
In PCI bus, no ADR pin will be shared with VAFC.
ASW0PU (Alternate Switch 0) will be added on MD pins to replace SW0 pin when VAFC is enabled by VAFCPU. ASW0PU will read-back in the same register bit as SW0. so the BIOS will not feel any modification in the reporting scheme. The chip will draw more power in suspend with VAFC.

5. Moved SW2, SW1/MCLK/XMCLK the two reserved pins (one obtained by killing CRTVDD)and TRWn on MVDD to prepare for SDRAM pin compatible option in the next Nordic that supports SDRAMs. Marked them as "reserved for SDRAM pin name". It looks that TRWn will not be needed, but it was moved on MVDD so that in case it is needed, it is available (in this case we'll need an extra switch for SDRAM support or at least to disable TRWn test mode effect).

6. Rearanged VAFC pins as pins R4 and R5 were used by VAFC - an error.

<u>Pinout Modifications done on 01/06/94 rev. 3.0</u>

1. Update Panel Data Pins based on Robin's Table.
2. Move MOD pin on a Panel Data Pin used in 18bit TFT.
3. Move OVRW on a 24bit Panel Data Pin.
4. Removed pin configuration for TV-OUT support with digital encoders. All pins marked TV... were removed. As a result, some pins like LLCLK. LFS. FPDE. FPVDCLK have one function now.
5. Removed WavePort pins, including AUDVDD: total 7 pins freed.
AUX2 and TVG muxed on EBROMn are no longer needed either.
6. Added one more CPU address pin: instead of HIMEM. Nordic has now HIMEM0 and HIMEM1, corresponding to CPU Address 24 and 25 or any other external decode of the high order CPU address bits. In the proccess of adding this pin, most CPU Interface pins on the right side were moved to the right by one pin.
7. Added CRTVDD supplying HSYNC. VSYNC, CSYNC. NTSC/PAL CRT and TV Interfaces output buffers.
8. EBROMn is shared with TVON, the power-on signal for AD720. So, in ISA bus with on chip BIOS support, AD720 power on cannot be directly controlled by Nordic.
9. Added pins for Analog TV Encoder support - AD720 and MC1377 :
    a. CSYNC = Composit Sync signal
    b. NTSC/PAL = 0 -> NTSC. 1-> PAL

c. EBROMn/TVON (see #8)

10. ASW0PU "Alternate Switch 0 Pull-Up" pull-up on pin 155 MD[26] was removed. as SW0 is now not multiplexed.

11. Demiltiplexed some of the pins previously multiplexed:

a. Pin 168 TWRn/RES4PMS is now RES4PMS. a pin reserved for synchronous DRAM support.

b. SUSPI/BLI/FCBLANKn is now SUSPI/BLI. not affected by the Feature Connector Enable Configuration

c. As a consequence of b. pin 97 is FCBLANKn (not muxed with anything) Please note that FCEVIDEOn is still muxed with SBYI/ACTI as before. as Standby Input and Keyborard Activity are less important functions in a system.

d. SW0 is no longer muxed on VCLK/FCDCLK. So VLK/FCDCLK is a pin but it moved to pin 103 in the PVDD area. to be consistent with FC power supply.

12. SW1. muxed on MCLK/XMCLK (pin197) became SW0. This is just a name change to have the switches placed in a more consistent manner.

13. SW1 is now muxed with a reserved pin: RES4CKE. pin 148. which before was totally reserved. The idea here is to go for pu/pd with SDRAM-s for SW2 and SW1. while keeping SW0 on MCLK even with SDRAM-s.

14. Pin 147 is Reserved. It is actually one of the WavePort pins. but placed in-between Memory and Panel VDD busses (on MVDD) for maximum flexibility. To acheive this. the pins on its right were moved to the right by one

FPVEE <BIAS> moved down to pin 98 (it was 105).

15. TRWn is now by itself on pin 96. on FPVDD.

16. The names of all flat panel data pins reflect the panel type table attached to this document.

Configuring Nordic-1M

If Terminator used specially assigned pins to configure the chip, **Nordic-1M will use some of the MD31:0 pins sampled on the Reset trailing edge,** similar to Mustang. The pad cells will have an active pull-down controlled by an **extended Reset** signal. The pull-down resistence will be about 75KOhm. An external pull-up less than 60KOhm is needed only if the option is used. In Suspend, the MD pads are inputs and the DRAM OEn is H so no DC current is drawn by the configuration h/w. An external resistor is needed only if the configuration pin has to be H at Reset. H/W configuration can be read via s/w too.

All h/w configuration latches are read accessible via I/O. The h/w configuration is latched on Reset trainling edge. (these latches were r/w, with s/w PU read, having

only read is enough).

We'll define the configuration pins to minimize the number of external resistors:

CPU Bus Configuration:

On MD18:16 & MD23
- -> no external PU = 32bits VESA VL bus at 33MHz or less bus clock
- -> MD16 PU = 32bits PCI bus with Burst <u>PU called "PCIPU"</u>
- -> MD17 PU = 16bits ISA bus <u>PU called "ISAPU"</u>
  ISAPU controls also the pin EBROMn/SLEEPIn
  In ISA bus this pin is EBROMn output; otherwise it is SLEEPIn input to be tied H if not needed.
- -> MD18PU = 32BIT VESA VL BUS AT 50MHZ BUS CLOCK <u>PU called "FVLPU"</u>
- -> MD23 PU = 32bits PCI bus no Burst <u>PU called "PCINBPU"</u>
  This is a reserved configuration in case burst does not work on a PCI bus.

Reads-back in SR22:

PCIPU on MD16 -> SR22[0]
ISAPU on MD17 -> SR22[1]
FVLPU on MD18 -> SR22[2]
PCINBPU on MD23 -> SR22[7]

Clock Configuration:

On MD19 (reads back on SR22[3])
- -> no external PU = internal MCLK and VCLK clocks, from the internal clock synthesizer
- -> MD19 PU = use external clocks from SW1/MCLK/XMCLK and OSC/XVCLK
  <u>PU called"XCLKPU"</u>

This PU is used for manufacturing test.

One reg. bit is needed if internal clock is selected to control outputing mclk on the MCLK/XMCLK pin = 1 if use internal clock then output it on SW0/MCLK/XMCLK pin. -> SR23[0] is this bit.

Memory Configuration: SRF[0]

No Pins, Use Two Register Bits -> 2CAS/2WE bit=H 2WE, bit=L 2CAS (default =L)
-> Symetric/Asymetric bit=H Asym., bit=L Sym. (default=L)

BIOS Support: (reads-back on SR22[6] for BIOS8PU)

Nordic-1M supports 16bit or 8bit BIOS and only in ISA bus configurations.
The BIOS is always supported when in ISA.

On MD22 -> no external PU = 16bit BIOS
-> external PU = 8bit BIOS (no MEMCS16n decode on C000)
This option is mostly for debug in case 16 bit BIOS range decode is not fast enough.
If the BIOS support is disabled, this configuration is don't care. Called "BIOS8PU".
Please note that Nordic-1M supports C000 BIOS only in ISA.

In a normal note-book the BIOS is at E000:0 or C000:0 but it is not supported by Nordic: so no external component is needed.

Sleep Address Selection:

On MD21: Sleep Address Selection (reads-back on SR22[5]).
-> No external PU = sleep at 3C3[0] - to be used when Nordic is on the Mother-board (most cases)
-> External PU = sleep at 46E8[3] - to be used when Nordic is an adapter (like a PCMCIA card and/or an adapter to turn desktop into an environment friendly PC) Called: S46PU
Note that Nordic-1M comes-up awake.

Feature Connector Configuration: (reads-back on SR24[7])
On MD25:
-> No external PU = no Feature Connector Support: outputs used only for FC are forced L. I/O are forced inputs. I/O and inputs are forced on the chip so that they do not take power if not connected externally.
-> External PU = All pins needed for FC including OVRW are active. Called: VAFCPU

~~On MD26: Alternate SW0 Selection (Reads-back on SR24[0] when SR24[7] is 1)~~

Total: 9 H/W PU configuration options.
In normal operation with FC enabled up to three external PU-s may be needed (bus type if not VL and FC).
In normal operation with FC disabled up to one external PU-s may be needed (bus type if not VL and FC)

Summary of PU/PD Configuration Pins and Register Bits where they can be read-back:

Cirrus Confidential Business Information

SR22 [7:0] H/W Configuration Read Register #1

Read only register. written on reset training edge by activating PU/PD on MDi pins. The PD-s are inside the pads active only during XReset (extended reset).

CL 152753

[7] 32bits PCI bus no Burst h/w PU configuration on MD23. It is called "PCINBPU" (read only bit).
This is a reserved configuration in case burst does not work on a PCI bus.

[6] Select 8 bit BIOS support (if 1). Otherwise 16 bit BIOS support if the BIOS support is at all enabled. PU on MD22.

[5] Resereved for Sleep Address Select: reads back S46PU (MD21).
1 = I/O address 46E8[3] as sleep address (needs external PU).
0 = I/O address 03C3[0] as sleep address (no external PU is needed).

[4] Reserved

[3] Select external clocks: MCLK and VCLK on SW0MCLK/XMCLK and OSC/XVCLK. (default: internal clock synthesizer, SW0 and OSC).
PU on XCLKPU on MD19.

0 = internal clocks with the pins used as inputs for panel type switch #1 and OSC. If SR23[4] and SR23[0] are 1, VCLK and SW0/MCLK/XMCLK, respectively become VCLK and MCLK outputs respectively, for test purposes (only if SR22[3]=0 MCLK can be seen, but VCLK can be seen even if SR22[3]=1 as long as SR23[4]=1)

1= SW0/MCLK/XMCLK and OSC/XVCLK become XMCLK and XVCLK inputs generating directly chip MCLK and VCLK without clock synthesizer. The clock synthesizers are powered down if SR22[3]=1.

[2] Select VESA VL Bus at 50MHz
1 = VESA VL bus at 50MHz selected (needs external PU on MD18)
0 = VESA VL bus at 50MHz not selected

[1] Select ISA Bus with PU on MD17.
1 = ISA bus selected (PU on MD17)
0 = ISA bus not selected (no PU on MD17)

Cirrus Confidential Business Information

[0] Select 32bit PCI Bus with PU on MD16.
1 = 32bit PCI bus selected (PU on MD16)
0 = 32 bit PCI bus not selected (no PU on MD16)

CL 152754

If [1:0] = 00 (i.e. no external PU on MD17:MD16) 32bit VESA VL bus is selected.
No more than one PU on MD17:MD16 should be used.

Note: In AVGA3, CF14 is "Enable Local BUs Address Latching Interface". Alpine does

not have this configuration bit. We need to do what was done in Alpine to get rid of this Configuration Pull-up/Pull-Down option. *CAN we? WAs this done. Dwarka?*

SR24 Panel Type Switches Register

[7] VAFCPU (VESA Advanced Feature Connector External Pull-up) Read-back. If external pull-up (the bit read-back 1) then all VAFC pins are configured for Video Port. Otherwise they have other functions or are disabled if no other function. (Disabled= the inputs are forced = the TTL and CMOST threshold controls are off & the outputs are tri-stated. It actually reads back what is latched on Reset on VAFCPU. VAFCPU is on MD25.

[2:0] SW2:0 pins read back (read only bits) ~~Please note that bit [0] reads back SW0 if VAFCPU (SR25[2]) is 0 and ASW0PU if VAFCPU is 1.ASW0PU is on MD26.~~

Configuration Register Bits needed:

Symetric/Asymetric DRAM -> SR8[7] in T1. CF13 in AVGA3
=> SRF[1] 0 = Symetric DRAM Default: 0
1 = Assymetric DRAM

This bit is read only in AVGA3 (reads back CF9), but it is writable in T1. It controlls MCLK frequency as an alternate to SR1F. We will use only SR1F to program MCLK frequency in Nordic. The default MCLK frequency will be 39.374MHz. corresponding to SR1F[5:0] = 18H=22D. This frequency fully meets the spec for -80 256Kx16 DRAM-s.

Two CAS or two WE DRAM -> pin 72 in T1. CF12 in AVGA3
=> SRF[0] 0 = 2 WE DRAM (Default:0)
1 = 2 CAS DRAM

16 or 32bit wide Video Memory (One or Two 256Kx16 DRAM-s or SDRAM-s)
-> not in T1, SRF[4:3] in AVGA3 (01=16bit. 10=32bit)
=> SRF[7,4:3] will be used in Nordic with the same encoding as in AVGA3. (11 is reserved for 64bit wide). Default: 10 (32bit). See Table next page. Please note that we have room to add other types of DRAM-s (SDRAM-s for instance).

In AVGA3 SR8[0] is "CS OUt to EEPROM" a feature Nordic does not Support. Nordic will not change the meaning of AVGA3 bits unless they are read only (like the pins or external PU read back).

SR8[2:0] are in T1 the read-back bits for SW2:0 input pins, but AVGA3 uses them for EEROM programming so we moved SW2:0 read back to SR24[2:0].

In T1 SR8[7] is symmetric/assymetric addressing... in Nordic this is SRF[1].

Two or one Bank 32bit wide DRAM

-> not in T1, SRF[7,4:3] in AVGA3 (00=8bit, 01=16bit, 10=32bit, 11=64bit. SRF[7] controls if 2MB of Video memory are available with 4x512kx8 -0- or with 4 256Kx16 DRAM-s, two banks -1-). *It seems that in AVGA3 to support 1MB with two 256Kx16 dram-s and 32bit wide bus, we select 32bit wide and SRF[7]... but I do not exactly know how this configuration is selected.*

=> SRF[7,4:3] in Nordic, will select the Video Memory Configuration slightly differently than in AVGA3 and Alpine. See the following table.

	SRF[7]	SRF[4]	SRF[3]	Memory Configuration
new	0	0	0	32bit, 2MB=4x512Kx8
	0	0	1	16bit, 512KB=1x256Kx16
	0	1	0	32bit, 1MB=2x256Kx16 (default)
	0	1	1	Reserved (64bit in Alpine)
new ?	1	0	0	32bit, 2MB=4x256Kx16, Two Banks
	1	0	1	Reserved
	1	1	0	Reserved
	1	1	1	Reserved

DRAM Timing select (short RAS or long RAS)

-> SRF[2] in T1 and AVGA3

=> SRF[2] in Nordic, default 1 = short RAS (3.5 L, 2.5H)

BIOS 48KB or 32KB -> not in T1; CF8 in AVGA3. In T1 SR8[4] is Suspend pin polarity. In Nordic Suspend pin is always active L (0 on pin puts the chip in Suspend).

=> SR23[2] in Nordic =0 32KB BIOS (default=0) C0000:C7FFF
=1 48KB BIOS : C0000:CCFFF

The initial BIOS code should be placed in the first 32KB starting at address C0000.

BIOS ROM ZERO* wait select in ISA and fast timing in VL bus

-> not in T1; CF1 in AVGA3

=> SR23[1] in Nordic =0 no ZERO wait state for BIOS in ISA

=1 ZERO wait state for BIOS in ISA. 8MHz bus

Output internal VCLK on VCLK/FCDCLK pin -> not in T1 or AVGA3

=> SR23[4] in Nordic =0
VCLK/FCDCLK is an output pin forced L
=1 either VCLK or FCDCLK is outputed on this pin. depending on VAFCPU (read back in SR24[7] value).

If SR24[7]=1 (Feature Connector pins enabled) than FCDCLK is outputed, otherwise, if

SR23[4]=1 than VCLK is outputed. VCLK out is needed only for manufacturing test.

Output internal MCLK on SW0/MCLK/XMCLK pin -> not in T1 or AVGA3.
=> SR23[0] =0 this pin is an input namely XMCLK if SR22[3]=1 (enable external clocks) and SW0 otherwise.
=1 output mclk

This pin will normally be used as SW0, except for manufaturing test where mclk and xmclk functions are needed.

SuspendI/BLI bit -> not in T1 or AVGA3.
=> SR23[5] in Nordic, 0= Suspend pin in Suspend/BLI pin (default)
1= BLI pin on Suspend/BLI pin

SBY/ACTI/FCEVIDEOn bit. -> not in T1 or AVGA3
=> SR23[6] in Nordic, 0= SBY on SBY/ACTI pin (default)
1= ACTI on SBY/ACTI pin

If SR24[7] = 1 (VAFC pins enabled) this pin is FCEVIDEOn independent of SR23[6].

VAFC Enable bit -> not in T1 or AVGA3.
(read only bit, actually reflecting the status of an external configuration PU: VAFCPU on MD25 latched at RESET or under S/W Control)
=> SR24[7] 0= VAFC Disabled at pins.
All VAFC dedicated pins are disabled to save power: inputs are disabled, outputs are L. I/O-s are disabled inputs (forced H) internally.
The shared pins are not in Feature Connector configuration.
1= VAFC Enabled at pins.
If any shared pins, they are now in Feature Connector Configuration.
All dedicated pins are active in this configuration.

Compressed Live Video (SashaPack) Enable bit -> not in T1 or AVGA3.
=> SR24[6] 0= disable (default)
1= enable

Cirrus Confidential Business Information

~~Audio In (from CS4215 to Nordic) Enable bit -> not in T1 or AVGA3.~~
~~=> SR24[5] 0= disable (force inputs inactive and outputs L~~

CL 152757

~~disable AI-FIFO and cycles~~
~~1= enable~~

~~Audio Out (from Nordic to CS4215) Enable bit -> not in T1 or AVGA3~~
~~=> SR24[6] 0= disable (as above but~~
~~AO-FIFO)~~
~~1= enable~~

Panel Type Switch Read-back -> SR8[2:0] in T1 or SR7[6:4], not in AVGA3
=> SR24[2:0] in Nordic, read only, writable on Reset or
under S/W Control (SR24[3])
When SR24[3] whose default is 0, transits from 1 to 0
SW2:0 pin value is latched into SR24[2:0 which cannot
be written by I/O.

Please **note** that **R9x[1:0]** TFT Panel type field of T1 was expanded to support 24bit TFT panels:

00 -> 9bit (333) (as in T1)
10 -> 12bit (444) (as in T1)
01 -> 18bit (666) [was x1 in T1 covering both 01 and 10 !!]
11 -> 24bit (888) -> new in Nordic

Notes:

Cirrus Confidential
Business Information

CL 152758

0.1 a. If 24bit TFT panel is selected, all related pins will be configured for it and it has priority over panel pins configuration settings for cross talk reduction enabled.
b. MOD is selected only with STN panels.
c. If 24bit TFT is not selected and VAFC is not selected, then SUSPST is selected on
G1/SUSPSTn/FCP[1] and SBYST is selected on G0/SBYSTn/FCP[0].
d. Panel Data unused output pins are forced 0 when not in use (depending on panel type)

SR25 Timers S/W Reset, H/W Config. #2 & Packed Modes Register

[7] The 8514A RAMDAC address is decoded as valid RAMDAC address in addition to the standard VGA RAMDAC address.
[6] Allows use of EBROMn as external 8514A RAMDAC
In this case it bahaves like 5428.
[5] Enable 1bit/pel packed mode
[4] Reserved for 2bit/pel packed mode
[3] Enable 4bit/pel packed mode
[2] Configure PCI bus Nordic module for Faster PCI write
[1] Reset Stand-by Timer
[0] Reset Backlight Timer

Other Register Related Topics
(See nordic-regs-rev#.lst where # is the latest one)

<u>BIOS SCratch Registers</u> #5 and #6 (2 extra required) (r/w)

SR28:SR29[7:0] are scratch registers.
Note that SR9, SRA, SR14 and SR15 are AVGA3 scratch registers which we keep in Nordic. Also there are 13 18 bit scratch registers in the RAMDAC is SR12[1]=1.

R9x - TFT Panel Data Format

----

[7] Reserved

[6:4 Select a 0 to 7 shift-clock delay from the internal character clock counter
(8 VCLK/Character Clock) to TFT panel HSYNC (LLCLK) signal -> as in T1.

[3:2] Panel size select:
00 -> 640x480
01 ->

[1:0] TFT Panel Type:

00 -> 9bit (333) (as in T1)
10 -> 12bit (444) (as in T1)
01 -> 18bit (666) [was x1 in T1 covering both 01 and 10 !!]
11 -> 24bit (888) -> new in Nordic

Registers that are Moved Relative to T1

CR1B[7,4,3,2] (R1B in schematics):

[7] Disable Text Cursor Blink in T1. Do we use this feature in T1?
Enable Blank End Extensions in AVGA3
[4] PCLKO (Panel Clock) no divide by two control in T1.
I am not sure this bit is actually needed in Nordic.
[3] IRQ Every Frame or Every Other Frame ? Is this bit used inT1?
[2] Not Used at All in T1.

I think that all these bits are not actually used. Are they needed ? -> Robin.

CR1C[7:0] & CR1D[7:0]

Are important LCD and Power Management Registers in T1 and Overlay Registers in Alpine. These registers are not in AVGA3, but because they are used in Alpine we may move them to **CR2C[7:0] and CR2D[7:0]**, which requires one more CRTC Index Register Bit. So:

CR1C -> CR2C & CR1D -> CR2D in Nordic relative to T1.

SRF[7]

In T1 is CRT Refresh Disable. This bit gates /FB/RWC[5] in FB sheet 6. In AVGA3 is DRAM Two Bank Configuration select bit.
I don't think that this bit is used in T1 ? -> Robin.

SR16

In T1 is used to control the pads for power and threshold. In AVGA3 is not used. In Alpine is CRT FIFO demand threshold and other things ([4:0] are used).

Move SR16 bits of T1 to SR20 to avoid contention with Alpine. Increase SRX Sequencer Register Index to 6 bits.

SR1A

In T1 is used as Test register. In AVGA3 and Alpine is Signature Generator Result Hi. Move T1 SR1A bits to SR21 if it is not replaced by a similar register in AVGA3.

Nordic Revision and Mask Codes

Nordic Microsoft ID CR2F: CD
Nordic Revision CR27 (4bits chip code, 7z , where z is BIOS rev. counting down): 2F
Nordic Mask Code CR25: 00
Nordic PCI ID: 1200H

-----$$$$-----